



2

NAVAL AEROSPACE MEDICAL RESEARCH LABORATORY
NAVAL AIR STATION, PENSACOLA, FL 32508-5700

NAMRL TECHNICAL MEMORANDUM 92-4

**COMPUTER SOFTWARE AND HARDWARE
TO DETERMINE CONTRAST SENSITIVITY
USING THREE METHODS: TRACKING,
LIMITS, AND CONSTANT STIMULI**

Donald J. Hatcher¹, Terry L. Devietti² and John A. D'Andrea¹

¹ Naval Aerospace Medical Research Laboratory
51 Hovey Road
Pensacola, Florida 32508-1046

² Central Washington University
Ellensburg, Washington 98926

DTIC
ELECTE
MAY 28 1993
S A D


Approved for public release; distribution unlimited.

93 5 27 02 2

93-12018



Reviewed and approved 15 Dec 92


A. J. MATECZUN, CAPT, MC USN
Commanding Officer



This research was sponsored by the Naval Medical Research and Development Command under work unit 62758N MM 58524.002-0010.

The views expressed in this article are those of the authors and do not reflect the official policy or position of the Department of the Navy, Department of Defense, nor the U.S. Government.

Trade names of materials and/or products of commercial or nongovernmental organizations are cited as needed for precision. These citations do not constitute official endorsement or approval of the use of such commercial materials and/or products.

Reproduction in whole or in part is permitted for any purpose of the United States Government.

Abstract

The software and hardware to obtain contrast sensitivity functions in a single session with monkeys are described. The data obtained from this program was used to examine the effect of microwave radiation on behavior in animals. The objective of collecting this data was to determine the safe operating conditions and levels of exposure for Navy personnel working with directed energy systems. The versatility of the software allows the program to be used in a variety of sensory systems with only minor programming modifications. It will permit absolute or difference threshold measurements with any one of three methods of threshold testing: (1) method of tracking, (2) method of limits, and (3) method of constant stimuli.

UNCLASSIFIED

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

Introduction

Contrast sensitivity functions (CSFs) are a fundamental description of spatial vision performance (Campbell and Green, 1965). During the past decade, spatial CSFs have become a widely used means to assess spatial visual performance in both humans and animals. Basically, it is the capability to perceive a target from its surroundings under dissimilar luminance conditions. Traditionally, sine wave gratings presented at different contrasts have been used to determine CSFs. Contrast, as pertaining to contrast sensitivity, can be defined as the difference in luminance of the background and the target. Spatial frequency (cycles/degree) is determined by the size of the target with small gratings representing a high spatial frequency and large gratings a low spatial frequency. By conducting tests at several spatial frequencies, a CSFs can be constructed to reflect the ability of the visual system to detect targets of different sizes and contrasts. While several studies have measured CSFs very effectively in rhesus monkeys (e. g., Boltz, Harwerth, & Smith, 1979; De Valois & Morgan, 1974; Jacobs, 1977; Williams, Boothe, Kiorpes, & Teller, 1981), the program and hardware described in this report were more productive for our requirement to evaluate visual function in rhesus monkeys (*Macaca mulatta*) during 1.5 h test sessions.

Several researchers (Blough 1958, 1966; Gourevitch, 1970; Rosenberger, 1970; Stebbins 1970) have established behavioral procedures to determine sensory thresholds in animals. Threshold determination procedures include 1) the method of tracking, 2) the method of limits, and 3) the method of constant stimuli. These procedures provided a basis for the development of our training procedure (DeVietti, D'Andrea, Hatcher, Reddix, submitted 1992), and for the software and hardware necessary to support the training procedure. In combination, these methods can be used to determine contrast sensitivity functions or other sensory thresholds using any one of the traditional threshold testing methods.

The following is an overview of the hardware and computer program developed to generate contrast displays on a very high resolution XYZ-axis display device (Tektronix 606B). The display consists of vertical sinusoidal gratings that can be computer controlled by the experimenter for varying degrees of contrast and spatial frequency. This program was developed to provide such visual stimuli to determine CSFs, record and store data, and identify threshold contrast sensitivity levels.

Hardware and Software Requirements

Certain hardware components are required for the program to function and evaluate contrast sensitivities. Figure 1 shows a block diagram of the complete setup required for proper operation. An IBM-compatible computer is used to run the program. A faster model, such as a 80486 25-MHz, will ensure accurate response-time measurements. Slower computers, for example the 80286, will suffice if accuracy of two or three hundredths of a second are not needed. A Tektronix XYZ monitor, model 606B, or compatible is used to display the gratings. To produce the gratings, a rectangle-pulse generator and a ramp-wave generator are used to produce inputs to the X and Y inputs of the XYZ monitor, respectively. A Metrybyte function generator model PCIP-SST is connected to the Z input to control the frequency and amplitude of the grating. In addition, a potentiometer is placed in series with the Z input of the monitor and the PCIP-SST to permit a wider range of amplitudes above and below the threshold level by effectively raising the zero voltage level to a positive voltage. To accurately measure the reaction time of responses, a Metrybyte CTM-5 timer is used. The timer card measures the amount of time it takes for the subject to make a left-lever response once the contrast has been displayed.

Program Operation

The program, TLC.EXE (appendix), was written and compiled using the Microsoft BASIC Professional Development System 7.1 (Microsoft Corp., 1987) and requires an IBM PC-compatible computer. A minimum disk space of 150K bytes is needed, and the program should be operated from a fixed disk. The program is "operator-input controlled" allowing the user to configure each experiment individually to the

CONTRAST SENSITIVITY

Block Diagram

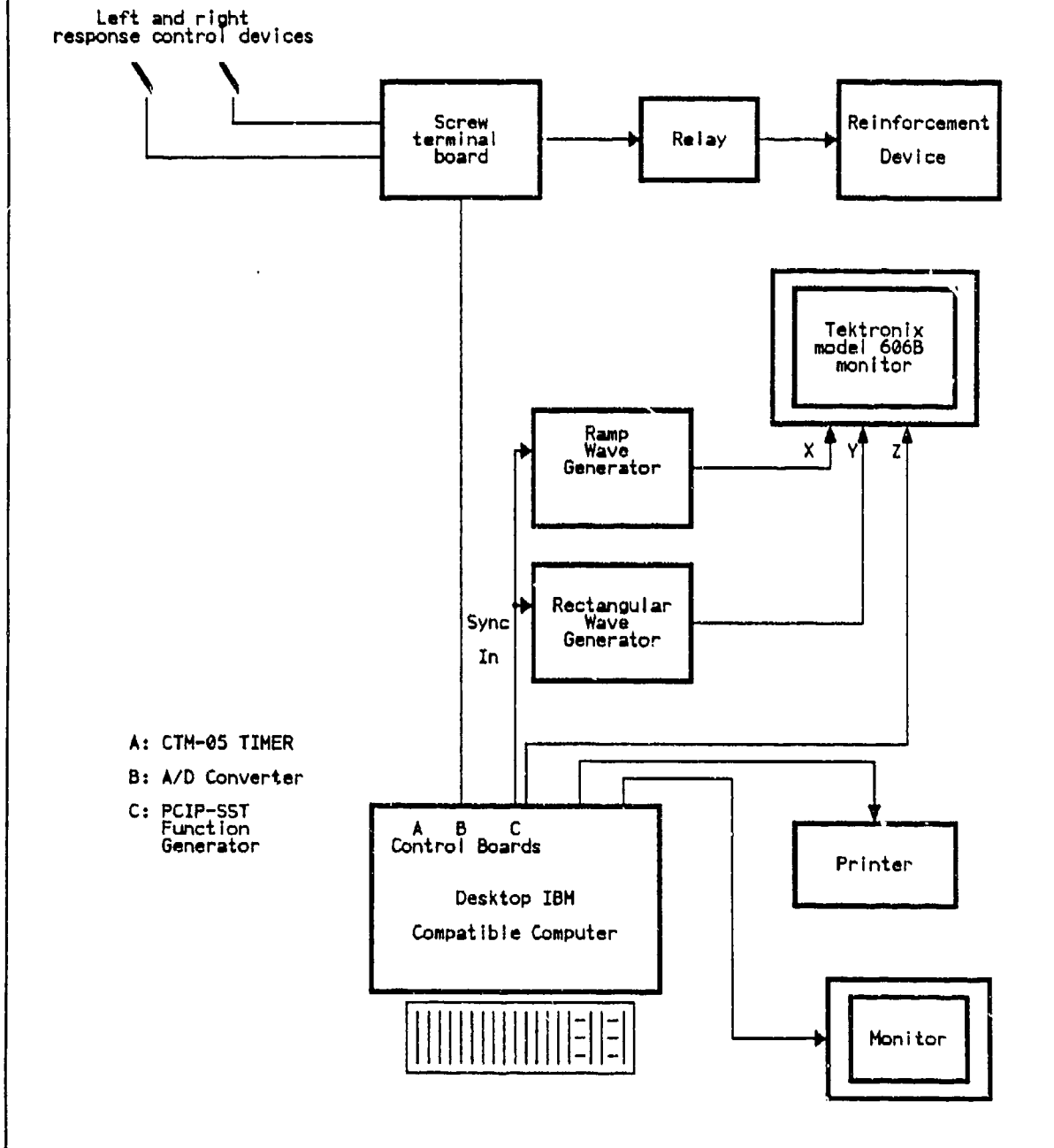


Figure 1. Block diagram of hardware.

method of testing, length of test, duration of contrast presented, timeout period (TO), spatial frequency, contrast value, variable interval (VI) and, if needed, catch trials. Certain variables, including VI, contrast intensity, timeout period, and spatial frequency, can be modified during the experiment by pressing the corresponding function key.

The program is divided into four stages. Stages 1, 2, and 3 are training stages. Stage 4 determines the contrast sensitivity threshold. A startup menu allows the experimenter to choose one of the stages (see Figure 2). The training stages are primarily designed to systematically teach the subject how to perform the desired task. The function of stage 1 is to familiarize the subject with the gratings and environmental surroundings. During this training, a contrast is always visible and reinforcements are supplied at variable intervals (VI) determined by the experimenter. The contrast can be displayed at fixed spatial frequencies and amplitudes, randomly selected spatial frequencies and amplitudes, or any combination of both.

10-30-1992 06:12:57

Contrast Sensitivity Procedures

1.....Stage 1 Training
2.....Stage 2 Training
3.....Stage 3 Training
4.....Stage 4 Procedures

Make Selection: ?

Figure 2. Program menu for choice of training or CSF testing.

Stage 2 introduces the subject to discrimination testing and is divided into two phases (Fig.3). Phase 1 of stage 2 requires the subject to make a left-lever response to obtain reinforcement. As in stage 1, spatial frequencies and amplitudes can be selected at random or remain at fixed values. Once the subject has mastered the reinforcement process, the stimulus is placed on a VI schedule (Phase 2). A left-lever response during a contrast produces a reinforcement, terminates the contrast, and selects the next VI. When the experimenter has determined that the subject is responding to the contrast, a second response control (right lever) is added. During this training process (Stage 3), a contrast is presented when the subject makes a right-lever response. Again, as in phase 2, a left-lever response delivers a reinforcement, removes the contrast, and determines the next VI. After the subject learns the process of pulling the right lever to present a contrast followed by a left-lever response for reinforcement, the gratings are placed on a VI schedule.

```

Stage 2 - Training Procedure

Enter file name.....: test.fil
Do you want random frequencies (Y/N): y
Do you want random intensities (Y/N): y
Session length in minutes.....: 60
Sleep Time in seconds.....: 4
Time-out in seconds.....: 10
Duration of Contrast.....: 4
Do you want catch trials (Y/N).....: n

You selected random frequencies and intensities

Frequency change rate in minutes....: .5
Intensity change rate in minutes....: .3

Information Correct? (Y/N)

```

Figure 3. Program menu for Stage 2 of training.

Stage 4 is the actual contrast sensitivity testing procedure (see Fig. 4). From this stage, the experimenter can determine the contrast sensitivities of the desired spatial frequencies by any of the three methods mentioned above. If the tracking method is selected, the program will automatically adjust the contrast values determined by a previously presented contrast. If the subject responds correctly to a presentation, the subsequent contrast will decrease by one step (0.06 log).

```

Stage 3 - Contrast Sensitivity Procedure

Enter file name.....: test.fil
(T)racking, (L)imits, (C)onstant....: t
Do you want random intensities (Y/N): N/A
Session length in minutes.....: 60
Sleep Time in seconds.....: 5
Time-out in seconds.....: 30
Duration of Contrast.....: 4
Do you want catch trials (Y/N).....: y

You selected the Tracking Mode

Enter valid frequency.....: 1230
Enter valid intensity.....: 2.505

Information Correct? (Y/N)

```

Figure 4. Program menu for contrast sensitivity testing.

When the subject fails to respond, the next contrast will increase by one step. This continues until the experimenter can determine a threshold based on whether the subject responds or does not respond to a contrast. Figure 5 is a flow chart for the tracking method used in the TLC.EXE program. Operating the

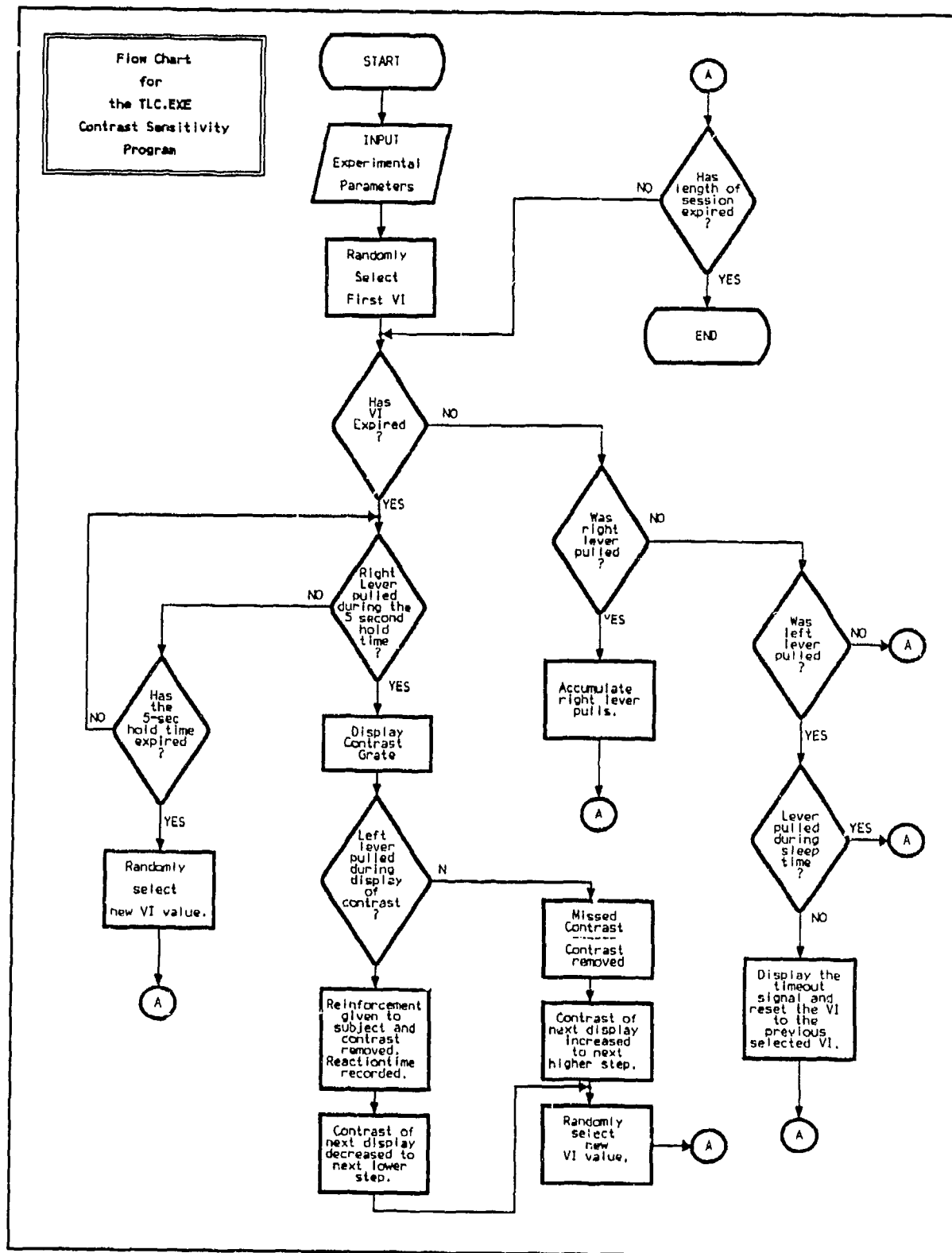


Figure 5. Flow chart of tracking method.

program in the limits mode allows the user to manually increase or decrease the contrast. Threshold levels are determined by recording the values to which the subject did and did not respond.

The last method, constant stimuli, allows the experimenter to dictate the amplitude values and the total number of times each contrast will be displayed. These values are selected at random until each contrast value has been displayed or the user terminates the test.

The duration of a manually presented contrast is indefinite and can be turned off by the subject making a left-lever response or by the experimenter pressing the space bar on the computer keyboard. In both cases, a reinforcement is delivered, the contrast is removed, and a new VI is started. The sleep time allows the subject to continue activating the left lever and not be penalized with a timeout. During the sleep time, the next interval is selected, and the program continues normally.

In all four stages, menu selections, as shown above, allow the user to input the file name, type of experiment (tracking, limits, constant stimuli), length of session, VI, timeout period, duration of stimulus, sleep time, catch trials, spatial frequency, and contrast amplitude. The VI is the average of all the variable intervals presented. Entering a VI of 20 would mean that the experimenter wants a contrast to be presented on an average of every 20 s. The intervals for each VI were selected using the formula developed by Fleshler and Hoffman (1962). The length of a timeout is entered in seconds. Each time the subject makes an incorrect response (left-lever response when a contrast is not being presented), the program automatically sends a 1-Hz, 0-volt signal to the monitor causing the display to flash on and off. This timeout (TO) period is used to reduce the amount of "guessing" by the subject that may occur at or near threshold levels. The duration of the stimulus is the length of time the contrast is displayed on the monitor. While the program is running, the experimenter can manually present a stimulus or deliver an immediate reinforcement to the subject.

To conclude that artifacts are not affecting subject responses, catch trials (.01 Vdc) may be entered into the random-interval selection process. These catch trials are presented exactly as a regular test grating and can help determine if any outside interference, such as noise, microwaves, or flicker are causing the subject to respond to the grating.

Spatial frequencies are determined by the distance of the eye to the target, size of the target, and the number of gratings displayed on the monitor. The number entered for a specific spatial frequency is the actual frequency of the PCIP-SST frequency generator the monitor requires to display a given spatial frequency. This will vary depending on the type of equipment being used and desired spatial frequencies. Data lines in the BASIC programming source code contain the frequency values required to display spatial frequencies of 1.5, 3.0, 6.0, 12.0, and 18.0 cycles/deg of visual angle with a target diameter of 3-cm and a distance from the subject of 100 cm using the hardware described previously.

To select the starting amplitude when using the tracking method, the experimenter enters a value that has been determined to be above the threshold level of the subject. Once a response is made to the grating, the next lower value is selected. The program uses contrast values between 0.2542 and 0.0035 in 0.06 log steps, which extend well above and below the normal threshold levels in both human (Vistech, 1988) and macaque (e.g., De Valois & Morgan, 1974).

Experimental Feedback: Screen, Printer, and Tones

The feedback the experimenter receives from the computer screen is shown in Figure 6 and is useful in determining certain statistics concerning the subject during the testing procedure. The PCIP-SST function generator is programmed to show the current output values being sent to the Tektronix 606B monitor, and is shown in the top portion of Figure 6. Frequency, amplitude, type of signal, and the dc offset value are

displayed and can be changed manually at any time during the experiment. The program updates every trial and displays the results on the computer screen. These data include the start time of the test session, current session time, current VI value and the remaining time until the VI expires, length of TOs, number of TOs, right-lever pulls with and without a contrast, and the total number of reinforcements received. In addition, the method of testing, spatial frequency, amplitude, and the average of all the VI's selected are displayed.

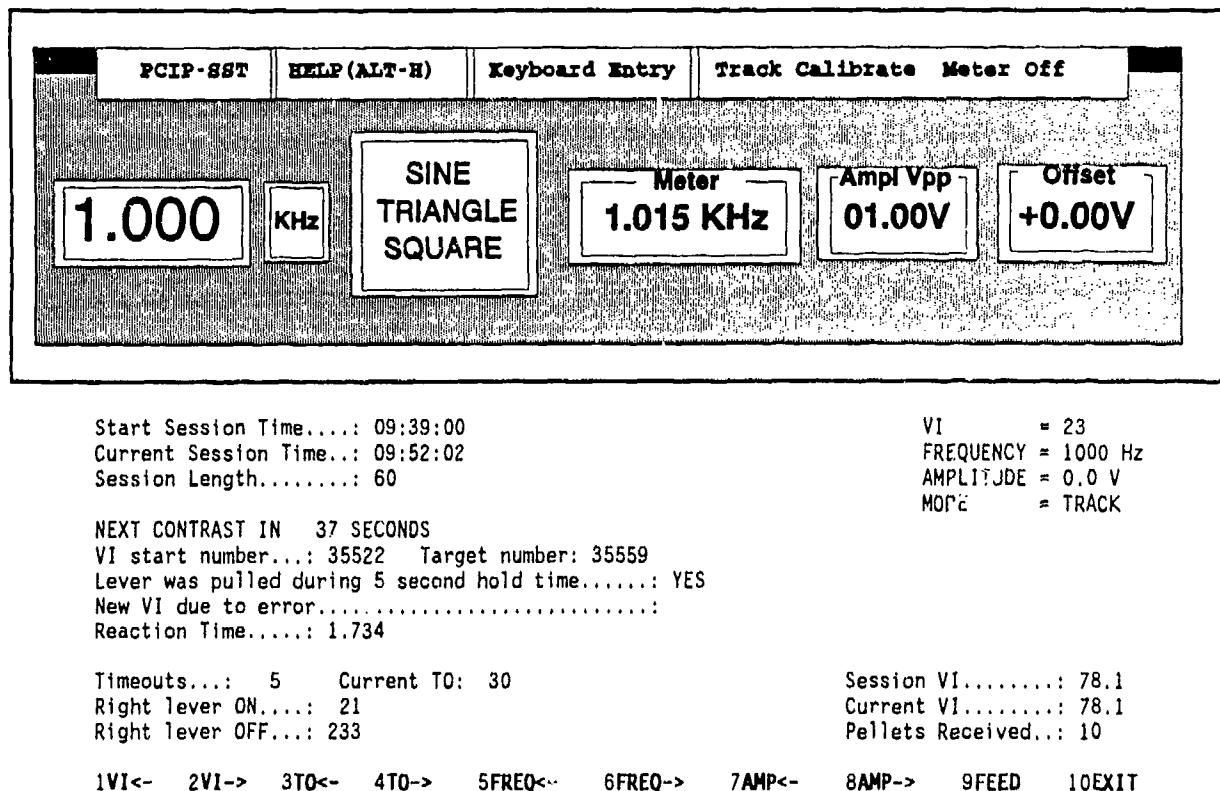


Figure 6. Illustration of on-line program feedback to the experimenter.

Output to a printer is updated following each trial showing the experimental method, reaction time, contrast value, spatial frequency, and a graph illustrating the progress from trial to trial. At the end of the experiment, the program compiles the data, performs all necessary calculations, and prints the results. When using the tracking method, the total number of trials and the number of trials that were responded to correctly at each contrast are printed (Fig. 7). A percentage of correct to incorrect responses is also indicated. All data are recorded to an ASCII file for further evaluation and future reference.

Audible tones are produced at the computer to indicate certain events such as right and left lever pulls, TOs, catch trials, and the expiration of a VI. By discerning the meaning of each tone, the experimenter can readily acknowledge the current status of the test. Not only does this feedback return the activity of the subject, it also provides the user with some assurance that the program is functioning properly.

Software requirements include the TLC.EXE program and the drivers needed to operate the PCIP-SST function generator. These PCIP-SST drivers come with the function generator and are loaded manually or automatically in the CONFIG.SYS file during the boot-up procedure of the computer. This software also

If changes are made to the source code, the program must be recompiled. A library, CTM5COM.LIB, containing the commands for the CTM-5 timer is supplied with the program. This library must be loaded from the command line (QBX /L CTM5COM.LIB) before the TLC.EXE file can be created.

SUMMARY

We developed a computer program (TLC.EXE) and training procedure¹ to determine contrast sensitivity functions in the monkey within a single session. The program controls all events and allows threshold testing by any one of three procedures (the method of constant stimuli, the method of limits, and the method of tracking). The program also provides both detailed on-line event-by-event monitoring and session summary results, and allows within-session changes in the procedure. Although the program was written initially to perform tests for contrast sensitivity, it can be used readily with minor modification to determine other sensory thresholds. It has been reconstructed in our laboratory to conduct hearing threshold experiments (AUDIO.EXE) in macaques. The output of the PCIP-SST function generator is used to change the frequency and amplitude of a signal connected to a speaker. Responses are made to a tone instead of a grating. Tracking is accomplished by decreasing or increasing the decibel level as the subject correctly or incorrectly responds to the tone. Because of the versatility of the TLC.EXE program, other sensory threshold experiments that utilize a sine, square, or triangle waveform from the function generator to present a stimulus to the subject can be readily accomplished.

¹ To receive a free copy of the software, send a self-addressed, stamped container with either a 3.5-in. or 5.25-in. formatted disk to the author.

STAGE 3

File - m1271622.fr2
04-10-1992
06:39:44

VI (sec)	Freq (Hz)	Am (V)	RT (sec)	SC	Relative Contrast Voltage	
					MIN	MAX
23- 42	1230	2.505	0.00	MVI		T
23- 11	1230	2.225	0.55			T
23- 35	1230	1.185	0.55			T
23- 40	1230	0.775	0.77			T
23- 43	0	0.000	0.00	C		T
23- 35	1230	0.555	0.74			T
23- 37	1230	0.325	0.00	MC		T
23- 5	1230	0.305	0.00	MC		T
23- 39	0	0.000	0.00	C		T
23- 9	1230	0.315	0.00	MC		T
23- 11	1230	0.325	1.63			T
23- 18	1230	0.325	0.00	MVI		T
23- 31	0	0.000	0.00	C		T
23- 4	1230	0.315	0.00	MC		T
23- 6	1230	0.325	0.00	MC		T
23- 37	0	0.000	0.00	C		T
23- 15	1230	0.445	1.40			T
23- 30	1230	0.445	0.00	MVI		T
23- 37	1230	0.325	0.00	MC		T
23- 8	0	0.000	0.00	C		T
23- 9	1230	0.445	1.77			T
23- 39	1230	0.325	1.74			T
23- 42	1230	0.315	0.00	MC		T
23- 23	0	0.000	0.00	C		T
23- 15	1230	0.325	2.32			T
23- 40	1230	0.315	1.21			T
23- 36	1230	0.305	0.00	MC		T
23- 4	0	0.000	0.00	C		T
23- 42	1230	0.305	0.00	MVI		T
23- 9	1230	0.315	2.05			T
23- 40	1230	0.305	0.00	MC		T
23- 23	1230	0.305	0.00	TD		T
23- 23	1230	0.315	0.00	MC		T
23- 41	0	0.000	0.00	C		T
23- 18	1230	0.325	0.00	MC		T

INITIAL SETTINGS:

Random Frequencies..... TRACKING
Random Amplitudes..... N/A

Session length (Min)..... 60

Frequency Rate Change (Min): 0
Amplitude Rate Change (Min): 0

Initial Frequency..... 1230
Initial Amplitude..... 2.505

Sleep Time (Seconds)..... 4
Time Out (Seconds)..... 40

Duration of Grating (Sec).... 4
Catch Trials (Y/N)..... Y

Figure 7. Real-time computer printout of an actual experimental test.

Frequency Tested: 1230 Hz

Contrast Sensitivity	Voltage	No. Presented	Correct Responses		Mean RT
			No.	% Correct	
164	0.305	3	0	0.0	0.00
143	0.315	6	2	33.3	1.63
124	0.325	7	3	42.9	1.89
108	0.445	3	3	100.0	1.68
94	0.555	1	1	100.0	0.74
72	0.775	1	1	100.0	0.77
47	1.185	1	1	100.0	0.55
27	2.225	1	1	100.0	0.55

Label Codes:

VI - Fixed Interval Freq - Frequency (Hz)
 Amp - Amplitude (Volts) RT - Reaction Time (sec)

Symbol Codes: (SC)

TD = Timeout MC = Missed Contrast MVI = Missed scheduled VI
 C = Catch Trial (No response) CR = Catch Trial (Response)
 EPC = Experimenter Presented Contrast

Procedure Codes:

T = Tracking L = Limits C = Contrast * = Other

Figure 7. (Continued)

References

- Blough, D.S. (1958). A method for obtaining psychophysical thresholds from the pigeon. *Journal of the Experimental Analysis of Behavior*, 1, 31-43.
- Blough, D.S. (1966). The study of animal sensory processes by operant methods. In W.K. Honig (Ed.), *Operant Behavior: Areas of Research and Application*, (pp. 345-379). New York: Appleton-Century-Crofts.
- Boltz, R.L., Harwerth, R.S., & Smith, E.L. (1979). Orientation anisotropy of visual stimuli in rhesus monkey: A behavioral study. *Science*, 205, 511-513.
- Campbell, F., & Green, D. (1965). Optical and retinal factors affecting visual resolution. *Journal of Physiology*, 181, 576-593.
- De Valois, R.L., & Morgan, H. (1974). Psychophysical studies of monkey vision--111. Spatial luminance contrast sensitivity tests of macaque and human observers. *Vision Research*, 14, 75-81.
- DeVietti, T.L., D'Andrea, J.A., Hatcher, D.J., & Reddix, M.D. A Training Procedure for Obtaining Contrast Sensitivity Functions Within a Single Session in Monkeys. Submitted to *Bulletin of the Psychonomic Society*.
- Fleshler, M. and Hoffman, H.S. (1962). A progression for generating variable-interval schedules. *Journal of the Experimental Analysis of Behavior*, 5 (4), 529-530.
- Gourevitch, G. (1970). Detectability of tones in quiet and in noise by rats and monkeys. In W. C. Stebbins (Ed.), *Animal Psychophysics*, (pp. 67-97). New York: Plenum Press.
- Jacobs, G.H. (1977). Visual capacities of the owl monkey (*Aotus trivirgatus*)-11. Spatial contrast sensitivity. *Vision Research*, 17, 821-825.
- Rosenberger, P.B. (1970). Response-adjusting stimulus intensity. In W.C. Stebbins, (Ed.), *Animal Psychophysics* (pp. 161-184). New York: Plenum Press.
- Stebbins, W.C. (1970). Studies of hearing and hearing loss in the monkey. In W. C. Stebbins (Ed.), *Animal Psychophysics* (pp. 41-66). New York: Plenum Press.
- Vistech Consultants, Inc. (1988). *Instruction Manual*. Available from Vistech Consultants, Inc., 1372 North Fairfield Road, Dayton, Ohio, 45432-2644.
- Williams, R.A., Boothe, R.G., Kiorpes, L., & Teller, D.A. (1981). Oblique effects in normally reared monkeys (*Macaca nemestrina*): Meridional variations in contrast sensitivity measured with operant techniques. *Vision Research*, 21, 1253-1266.

Other Related NAMRL Publications

None are applicable.

APPENDIX

BASIC Source Code for TLC.EXE

' TLC.BAS

' Source Code Written By: Donald J. Hatcher
' Naval Aerospace Medical Research Laboratory
' Pensacola, Florida

' Last Update: 02/04/92

' This program uses the PCIP-SST Function Generator, CTM-TIMER, and Acquisition Board

DECLARE SUB initCTM5 (BSA%)
DECLARE SUB zeroCTM51 (BSA%)
DECLARE SUB readCTM51 (ct!(), BSA%)
DECLARE SUB loadvi ()

COMMON file\$, fa\$, fb\$, fd\$, ov1\$, ov2\$, ov3\$
CLS
CLEAR , , 10000
OPEN "\$SST" FOR OUTPUT AS #1 ' Open Function Generator PCIP-SST
CLS

DIM SHARED VI(0 TO 144, 1 TO 30) AS INTEGER

DIM ct(10), freq\$(60), Amp\$(3, 80), VIS(16)
DIM Tamp\$(1000), Tamp(1000), TampC(1000), rt(1000)
DIM Lamp\$(1000), Lamp(1000), LampC(1000)
DIM count(1000)
DIM Voltage(32)
DIM Sensitivity(32)

RANDOMIZE TIMER

menu\$ = "Contrast Sensitivity Procedures"
make\$ = "Make Selection: ?"
center = 40 - (LEN(menu\$) / 2)
make = 40 - (LEN(make\$) / 2)
a\$ = ""
averagepellets = 0

OUT 779, 136
OUT 777, 0
BSA% = &H310

CALL initCTM5(BSA%)
CALL zeroCTM51(BSA%)
CALL readCTM51(ct(), BSA%)

KEY 1, "VI<-"
KEY 2, "VI>"
KEY 3, "TO<-"
KEY 4, "TO>"
KEY 5, "FREQ<-"
KEY 6, "FREQ>"
KEY 7, "AMP<-"
KEY 8, "AMP>"
KEY 9, "FEED"
KEY 10, "EXIT"

ON KEY(1) GOSUB DecreaseVI
ON KEY(2) GOSUB IncreaseVI
ON KEY(3) GOSUB DecreaseTimeOut
ON KEY(4) GOSUB IncreaseTimeOut
ON KEY(5) GOSUB DecreaseFrequency
ON KEY(6) GOSUB IncreaseFrequency
ON KEY(7) GOSUB DecreaseAmplitude
ON KEY(8) GOSUB IncreaseAmplitude
ON KEY(9) GOSUB Feed
ON KEY(10) GOSUB endphase1

VIA = 16
FOR c = 1 TO VIA
 READ VIS(c)
NEXT c

freqs = 5
FOR loadfreq = 1 TO freqs
 READ freq\$(loadfreq)
NEXT loadfreq

FOR Table = 1 TO 2
 FOR loadamp = 1 TO 18
 READ Amp\$(Table, loadamp)
 NEXT loadamp
NEXT Table
Table = 3

amps = 42
FOR loadamp = 1 TO amps
 READ Amp\$(Table, loadamp)
NEXT loadamp

CALL loadvi

contrasts = 32
FOR loadcontrast = 1 TO contrasts

```

    READ Voltage(loadcontrast)
NEXT loadcontrast

```

```

FOR loadcontrast = 1 TO contrasts
    READ Sensitivity(loadcontrast)
NEXT loadcontrast

```

```

RESTORE

```

```

ON ERROR GOTO errcheck
LOCATE 10, 20: PRINT "Checking Status of Printer....."
LPRINT CHR$(255);
LPRINT CHR$(27); CHR$(15);
WIDTH "LPT1:", 130

```

```

mainmenu:

```

```

    CLS
    ss = 0
    ERASE Tamp, TampC, rt, Lamp, LampC
    LleverON = 0
    PullSleep = 0
    fa$ = "\    \    \    \    \    \    \    \    \    \    \"
    fb$ = "\    \    \    \    \    \    \    \    \    \"
    fd$ = "####\####  ####  ##.###  ###.###  \    \"
    \"
    fre$ = "FREQUENCY = \    \ Hz \"
    ampl$ = "AMPLITUDE = \    \ Volts\"
    vi$ = "VI      = ### \"
    mode$ = "MODE      = \    \"
    previous = 0
    LOCATE 2, 2: PRINT DATES$
    LOCATE 4, center
    PRINT menu$
    LOCATE 6, 23: PRINT "1.....Stage 0 Training\"
    LOCATE 7, 23: PRINT "2.....Stage 1 Training\"
    LOCATE 8, 23: PRINT "3.....Stage 2 Training\"
    LOCATE 9, 23: PRINT "4.....Stage 3 Procedures\"
    LOCATE 10, 23: PRINT "X.....Quit Program\"
    LOCATE 12, make: PRINT make$
    selection = 0

```

```

TestKey:

```

```

    WHILE selection = 0
        a$ = INKEY$
        LOCATE 2, 68: PRINT TIMES$
        IF a$ = "" THEN GOTO TestKey
        selection = INSTR("1234Xx", a$)
    WEND
    ON selection GOSUB Stage0, Stage1, Stage2, Stage2, quit, quit
    GOTO mainmenu

```

Stage0:

```
FOR keyn = 1 TO 2
  KEY(keyn) ON
NEXT keyn
```

```
FOR keyn = 5 TO 10
  KEY(keyn) ON
NEXT keyn
```

```
stage$ = "Stage 0"
combo = 1
Fed = 0
phase = 0
testphase = 1
GOSUB EntryScreen
CLS
PRINT #1, "SHOW ME ON"
KEY ON
```

```
PRINT #1, "AMP=" + Amp$(Table, t)
PRINT #1, "FREQ=" + freq$(n)
PRINT #1, "ME ON"
ON TIMER(length) GOSUB endphase1
TIMER ON
LOCATE 9, 1: PRINT "Starting Time.....: "; TIMES$
LOCATE 10, 1: PRINT "Current Time.....: "
LOCATE 11, 1: PRINT "Session Length.....: "; SessionLength
LOCATE 13, 1: PRINT "Number pellets received..: "
fcr1 = TIMER + FCR
icr1 = TIMER + ICR
GOSUB SelectVI
LOCATE 15, 1: PRINT USING "First Pellet in ### Seconds"; V1set
pellet1 = TIMER + V1set
LOCATE 9, 54: PRINT USING wvi$; v
LOCATE 10, 54: PRINT USING fre$; freq$(n)
LOCATE 11, 54: PRINT USING ampl$; Amp$(Table, t)
LOCATE 12, 54: PRINT USING mode$; "Stage 0"
```

Bp0:

DO

```
LOCATE 10, 28: PRINT TIMES$
```

```
fcr2 = TIMER
icr2 = TIMER
pellet2 = TIMER
```

```
IF INT(fcr2) >= INT(fcr1) THEN
  GOSUB FreqChange
  LOCATE 10, 54: PRINT USING fre$; freq$(n)
END IF
IF INT(icr2) >= INT(icr1) THEN
```

```

    GOSUB IntensityChange
    LOCATE 11, 54: PRINT USING amp!$; Amp$(Table, t)
END IF

IF INT(pellet2) >= INT(pellet1) THEN
    OUT 777, 4          ' Turn on Pellet Feeder
    FOR x = 1 TO 3500: NEXT x      ' Delay for feeder operation
    OUT 777, 0          ' Turn off Pellet Feeder
    Fed = Fed + 1
    LOCATE 13, 28: PRINT USING "###"; Fed
    GOSUB SelectVI
    LOCATE 15, 1: PRINT USING "Next Pellet in ### Seconds "; V!set
    pellet1 = TIMER + V!set
END IF

```

LOOP

Stage1:

```

    FOR keyn = 1 TO 2
        KEY(keyn) ON
    NEXT keyn

```

```

    FOR keyn = 5 TO 10
        KEY(keyn) ON
    NEXT keyn

```

```

stage$ = "Stage 1"
st = 1
Table = 1
phase = 2
testphase = 2
GOSUB EntryScreen
GOSUB PrintHeader
LPRINT USING fa$; "VI"; "Freq"; "Amp"; "RT"; "SC"; "Relative Contrast Voltage"
LPRINT USING fb$; "(sec)"; "(Hz)"; "(V)"; "(sec)"; "MIN " + STRING$(49, " - ") + " MAX"
LPRINT STRING$(130, "-")
LPRINT

```

```

WRITE #2, st, av
PRINT #1, "SHOW ME ON"
KEY ON
GOSUB Phase2a
PRINT #1, "ME OFF"
GOTO endphase1

```

Phase2a:

```

CLS
phase = 1
Table = 1
bp1 = 1

```

```

ss = 1
ResetVIsset = 0
contrastON = 0
ContrastOFF = 0
totaltimeouts = 0
extendVI = 0
fcr1 = TIMER + FCR
icr1 = TIMER + ICR
pellet1 = TIMER + VIsset
first = TIMER
first2 = TIMER
pulls = 0
Changed = 0
Fed = 0
fd = 1
av = 18
VIon = 0
sh = 1
LOCATE 19, 1
PRINT USING "Reaction Time.....: ###.###"; ReactionTime

```

Bp:

```

PRINT #1, "AMP=" + Amp$(Table, t)
PRINT #1, "FREQ=" + freq$(n)
PRINT #1, "ME ON"
LOCATE 9, 54: PRINT USING wv$; v
LOCATE 10, 54: PRINT USING fre$; freq$(n)
LOCATE 11, 54: PRINT USING ampl$; Amp$(Table, t)
LOCATE 12, 54: PRINT USING mode$; "Stage 1"
CALL zeroCTM51(BSA%)
ON TIMER(length) GOSUB endphase1
TIMER ON
LOCATE 9, 1: PRINT "Start Session Time   "; TIMES$
LOCATE 11, 1: PRINT "Session Length....."; SessionLength

```

BeginProcedure:

DO

```

LOCATE 10, 1: PRINT "Current Session Time..: "; TIMES$
LOCATE 12, 1: PRINT SPACES$(30)

```

IF phase = 2 THEN

LOCATE 12, 1: PRINT STRING\$(50, " ")

LOCATE 13, 1: PRINT USING "NEXT CONTRAST IN ### SECONDS "; VIsset

END IF

IF VIon > 0 THEN

LOCATE 14, 1: PRINT "VI start number....: "; INT(checkpull), "Target number: "; INT(pellet1)

END IF

fcr2 = TIMER

icr2 = TIMER

```

pellet2 = TIMER
checkpull = TIMER
pelletavetimer = TIMER
IF v > 0 THEN
    ctime = TIMER
    tctime = TIMER
END IF

```

```

IF INT(fcr2) >= INT(fcr1) AND sh = 0 AND combo = 1 THEN GOSUB FreqChange
IF INT(icer2) >= INT(icer1) AND sh = 0 AND combo = 1 THEN GOSUB IntensityChange
IF INT(fcr2) >= INT(fcr1) AND sh = 0 AND combo = 2 THEN GOSUB FreqChange
IF INT(icer2) >= INT(icer1) AND sh = 0 AND combo = 3 THEN GOSUB IntensityChange

```

```

lever1 = INP(768)
IF lever1 > 2 AND Vlon = 0 THEN
    GOSUB Graph
    GOSUB GetReinforcement
    GOSUB TestPulls
    sh = 0
END IF

```

```

IF lever1 > 2 AND Vlon = 2 THEN
    ContrastOFF = ContrastOFF + 1
    SOUND 400, .4
    GOSUB TestPulls
    checkpull = TIMER
    GOSUB Setok
END IF

```

```

IF Vlon = 1 THEN
    GOSUB GetReinforcement
    sh = 0
    Vlon = 2
END IF

```

LeverHold2:

```

IF INT(pellet2) >= INT(pellet1) AND Vlon = 2 THEN
    stage = 3
    GOSUB ShowGrad
    sh = 1
END IF

```

```

IF Vlon = 3 AND lever1 > 2 THEN Vlon = 1

```

LOOP

Stage2:

```

FOR keyn = 1 TO 10

```

KEY(keyn) ON
NEXT keyn

IF a\$ = "3" THEN stage\$ = "Stage 2" ELSE stage\$ = "Stage 3"

st = 2

IF stage\$ = "Stage 3" THEN st = 3

IF st = 2 THEN Table = 2

IF st = 3 THEN Table = 3

phase = 4

testphase = 0

LleverON = 0

LleverOff = 0

RleverON = 0

RLeverOFF = 0

VITM = 0

GOSUB EntryScreen

GOSUB PrintHeader

LPRINT USING fa\$; "VI"; "Freq"; "Amp"; "RT"; "SC"; "Relative Contrast Voltage"

LPRINT USING fb\$; "(sec) "; "(Hz)"; "(V)"; "(sec)"; "MIN " + STRING\$(49, "-") + " MAX"

LPRINT STRING\$(130, "-")

LPRINT

IF y1\$ = "T" THEN

ActiveMode\$ = "TRACK"

LOCATE 12, 54: PRINT USING mode\$; ActiveMode\$

END IF

KEY OFF

KEY 3, "TO <"

KEY 4, "TO >"

ON KEY(3) GOSUB DecreaseTimeOut

ON KEY(4) GOSUB IncreaseTimeOut

KEY(3) ON

KEY(4) ON

KEY ON

IF y1\$ = "L" THEN

ActiveMode\$ = "LIMITS"

LOCATE 12, 54: PRINT USING mode\$; ActiveMode\$

END IF

IF stage\$ = "Stage 3" THEN

IF y1\$ = "C" THEN

ActiveMode\$ = "CONSTANT"

p = 0

SOUND 470, .4

LOCATE 15, 1: PRINT SPACES\$(70)

LOCATE 15, 20: INPUT "Number of Amplitude Values: "; av

FOR novalues = 1 TO av

SOUND 470, .4

LOCATE 15, 20: PRINT "Amplitude Value number "; novalues; " : "; SPACES\$(10)

```

        LOCATE 15, 47: INPUT Amp$(Table, novalues)
        LOCATE 20, 2: PRINT "Amplitudes Selected: "
        LOCATE 20, 24
        p = p + 1
        FOR pp = 1 TO p
            PRINT Amp$(Table, pp); " ";
        NEXT pp
    NEXT novalues

    t = INT(RND * av) + 1
Sfq:    SOUND 470, .4
        LOCATE 17, 17: INPUT "Number of presentations per Value.....: ", np
        SOUND 470, .4
        LOCATE 18, 17: INPUT "Enter valid frequency.....: ", frq$
        FOR c = 1 TO (av * np)
            count(c) = 0
        NEXT c
        FOR n = 1 TO freqs
            IF frq$ = freq$(n) THEN
                freq$ = frq$
                TrackingFrequency$ = freq$
                GOTO Skipsfq
            END IF
        NEXT n
        BEEP
        GOTO Sfq
Skipsfq:
        LOCATE 17, 17: PRINT SPACES$(60)
        END IF
    END IF
    WRITE #2, st, av

    PRINT #1, "SHOW ME ON"
    KEY ON
    ON combo GOSUB Phase3a, Phase3a, Phase3a, Phase3a
    PRINT #1, "ME OFF"
    GOTO endphase1

```

Phase3a:

```

    CLS
    response$ = "N"

    PRINT #1, "AMP=20"
    PRINT #1, "FREQ=" + freq$(n)

    display = 1

    PRINT #1, "AMP = 0 FREQ = 1"

```

Phase3b:

```

    VIon = 0

```



```

phase = 3
CLS
LOCATE 10, 30: INPUT "Enter Valid VI: "; v

FOR c = 1 TO VIA
  IF VIS(c) = v THEN GOTO Phase3acont
NEXT c
BEEP
LOCATE 12, 10
PRINT "Not a Valid VI - Enter 0, 4, 6, 8, 10, 12, 16, 20, 22, 23, 24, 26, 30, 36, 144"
GOTO Phase3b

```

Phase3acont:

```

CLS
IF v >= 4 THEN GOSUB InitVI
ManualGrates = 0
fedmanual = 0
contrastON = 0
ContrastOFF = 0
bp1 = 0
totaltimeouts = 0
extendVI = 0
trial = 0
Cat = 0
catchtrial = 0
getgrad = 0
sh = 0
pulls = 0
Changed = 0
Fed = 0
LOCATE 19, 1: PRINT USING "Reaction Time.....: ###.####"; ReactionTime
SessionLengthSeconds = 0
SessionLengthSeconds = TIMER

```

Bp2:

```

PRINT #1, "AMP=0"
PRINT #1, "FREQ=" + freq$(n)
LOCATE 9, 54: PRINT USING vvi$; v
LOCATE 10, 54: PRINT USING fre$; freq$(n)
LOCATE 11, 54: PRINT USING ampl$; "0"
IF st = 2 THEN
  ActiveMode$ = "Stage 2"
  LOCATE 12, 54: PRINT USING mode$; ActiveMode$
END IF
CALL zeroCTM51(BSA%)
IF y1$ <> "C" THEN ON TIMER(length) GOSUB endphase1
TIMER ON
LOCATE 9, 1: PRINT "Start Session Time.....: "; TIME$
LOCATE 11, 1: PRINT "Session Length.....: "; SessionLength

```

```

WHILE l < 10
  PRINT #1, "AMP=20"
  PRINT #1, "FREQ =" + freq$(n)
  l = INP(768)
WEND

WHILE l > 2
  l = INP(768)
  aaa$ = INKEY$
  IF aaa$ = CHR$(13) THEN l = 0
WEND

GOSUB FeedPellet
firstpellet = 1
delay = TIMER + SleepTime

IF UCASE$(trials$) = "Y" THEN
  SELECT CASE v
    CASE 6
      catch = 14
    CASE 8
      catch = 12
    CASE 10
      catch = 11
    CASE 12
      catch = 10
    CASE 16
      catch = 8
    CASE 20
      catch = 6
    CASE 22
      catch = 5
    CASE 23
      catch = 5
    CASE 24
      catch = 5
    CASE 26
      catch = 4
    CASE 30
      catch = 4
    CASE 36
      catch = 3
  END SELECT
END IF

GOSUB InitVI

count = 1
averagepellets = 1
fcr1 = TIMER + FCR
icr1 = TIMER + ICR
pellet1 = TIMER + VIsset

```

```

first = TIMER
first2 = TIMER
show = TIMER + duration
IF y1$ = "C" THEN VIon = 1

```

BeginProcedure2:

DO

```

LOCATE 10, 1: PRINT "Current Session Time..: "; TIME$; SPACES$(20)
lever1 = INP(768)
lever2 = INP(770)
IF lever2 > 2 AND VIon = 0 THEN
    stage = 3
    lever2 = 0
    GOSUB ShowGrad
    sh = 1
END IF
LOCATE 12, 1: PRINT STRING$(50, " ")
LOCATE 13, 1: PRINT USING "NEXT CONTRAST IN ### SECONDS "; VIsset

```

```

IF TLC$ = "C" THEN
    total = 0
    FOR c = 1 TO av
        total = total + count(c)
        IF total = (av * np) + 1 THEN GOTO endphase1
    NEXT c
END IF

```

```

fcr2 = TIMER
icr2 = TIMER
pellet2 = TIMER
show2 = TIMER
ctime = TIMER
tctime = TIMER
delay2 = TIMER
pelletavetimer = TIMER
checkpull2 = TIMER

```

```

LOCATE 15, 72: PRINT " "

```

```

IF INT(show2) >= INT(show) AND VIon = 2 THEN

```

RFHR:

```

IF fedmanual = 0 THEN
    missed$ = "MC"
    ReactionTime = 0
    miss = 1
    mc = 1
ELSE
    missed$ = "EPC"

```

```

    fedmanual = 0
    IF mc = 0 THEN miss = 0
    IF mc = 1 THEN miss = 1
END IF
'miss = 1
ctrial = ctrial + 1

WRITE #2, v, Viset, freq$(n), Amp$(Table, t), ReactionTime, Totalex, missed$, TLC$
Amp = VAL(Amp$(Table, t))
freq = VAL(freq$(n))
IF missed$ = "EPC" THEN Amp = 20
IF UCASE$(TLC$) = "T" THEN GOSUB Tgraph
IF UCASE$(TLC$) = "L" THEN GOSUB Lgraph
IF UCASE$(TLC$) = "C" THEN GOSUB Cgraph
IF LEN(TLC$) < 1 THEN GOSUB Graph
LPRINT USING fd$; v; "-"; Viset; freq; Amp; ReactionTime; missed$; t$
GOSUB NewVI2
PRINT #1, "AMP=0"
PRINT #1, "FREQ="; freq$(n)
Vion = 1
missed$ = " "
END IF
IF delay2 >= delay THEN
    levercheck = INP(768)
    IF levercheck > 2 AND Vion = 2 THEN
        SOUND 1600, .4
        GOSUB GetReinforcement
        delay = TIMER + SleepTime
        sh = 0
        IF phase = 4 THEN
            Vion = 1
            IF st <> 2 THEN GOSUB FreqChange
            IF phase = 3 AND st <> 2 THEN GOSUB IntensityChange
        END IF
    END IF
END IF
IF INT(pellet2) >= INT(pellet1) AND Vion = 1 THEN
    GOSUB ShowGrad
    sh = 1
END IF
IF INT(fcr2) >= INT(fcr1) AND sh = 0 AND combo = 1 AND st = 2 AND Vion <> 2 THEN
    GOSUB FreqChange
END IF
IF INT(icer2) >= INT(icer1) AND sh = 0 AND combo = 1 AND st = 2 AND Vion <> 2 THEN
    GOSUB IntensityChange
END IF
IF INT(fcr2) >= INT(fcr1) AND sh = 0 AND combo = 2 AND st = 2 AND Vion <> 2 THEN
    GOSUB FreqChange
END IF
IF INT(icer2) >= INT(icer1) AND sh = 0 AND combo = 3 AND st = 2 AND Vion <> 2 THEN
    GOSUB IntensityChange
END IF

```

```

IF delay2 >= delay THEN
  IF VIon < 3 AND lever1 > 2 THEN GOSUB CheckLeft
END IF
IF st >= 2 THEN LOCATE 14, 1: PRINT "VI start number...: "; INT(checkpull2), "Target number:";
INT(pellet1)
IF VIon = 2 AND lever2 > 2 THEN GOSUB CheckRight
IF VIon = 3 AND lever2 > 2 AND stage = 3 THEN GOSUB CheckRight
IF VIon = 3 AND lever2 > 2 AND phase = 3 THEN GOSUB CheckRight
IF VIon = 1 AND lever2 > 2 AND phase = 4 THEN GOSUB CheckRightOFF
IF VIon = 3 AND lever1 > 2 THEN
  SOUND 1600, .4
  GOSUB GetReinforcement
  sh = 0
  IF phase = 3 THEN
    VIon = 0
    IF st <> 2 THEN GOSUB FreqChange
    IF phase = 3 AND st <> 2 THEN GOSUB IntensityChange
  END IF
END IF

```

LOOP

TuneUp:

```

freq = VAL(freq$(n))
freq = freq + 5
freq$(n) = STR$(freq)
IF display = 1 THEN LOCATE 10, 54: PRINT USING fre$; freq$(n)
RETURN

```

TuneDown:

```

freq = VAL(freq$(n))
freq = freq - 5
IF freq < 10 THEN freq = 5
freq$(n) = STR$(freq)
IF display = 1 THEN LOCATE 10, 54: PRINT USING fre$; freq$(n)
RETURN

```

Limit:

```

TLC$ = "L"
ActiveMode$ = "LIMIT"
LOCATE 12, 54: PRINT USING mode$; ActiveMode$
KEY 3, "TRACK"
KEY 4, " "
ON KEY(3) GOSUB Track
KEY(3) ON
KEY(4) OFF

```

RETURN

Track:

```
TLC$ = "T"  
ActiveMode$ = "TRACK"  
LOCATE 12, 54: PRINT USING mode$; ActiveMode$  
KEY 3, "LIMIT"  
KEY 4, " "  
ON KEY(3) GOSUB Limit  
KEY(3) ON  
KEY(4) OFF
```

RETURN

CheckLeft:

CheckLeft2:

SOUND 400, .4

IF firstpellet = 1 THEN

firstpellet = 0

RETURN

END IF

IF delay2 < delay AND SleepTime > 0 THEN

ps = 1

PullSleep = PullSleep + 1

RETURN

END IF

LleverOff = LleverOff + 1

LOCATE 21, 1: PRINT USING "Timeouts.: ### Current TO: ###"; LleverOff; Timeout

Tout = Timeout + TIMER

CheckTimeout = TIMER

TimeOutRoutine:

WHILE CheckTimeout < Tout

CheckTimeout = TIMER

c = INP(768)

IF c > 2 THEN Tout = TIMER + Timeout

chk = 1

LOCATE 9, 40: PRINT CHR\$(1)

LOCATE 10, 1: PRINT "Current Session Time.: "; TIME\$

checkpull = TIMER

IF Vlon > 0 THEN

LOCATE 14, 1: PRINT "VI start number...: "; INT(checkpull), "Target number: "; INT(pellet1)

END IF

PRINT #1, "FREQ=1 AMP=0 ME ON"

WEND

IF Vlon < 3 THEN GOSUB TestPulls

PRINT #1, "AMP=0"

PRINT #1, "FREQ="; freq\$(n)

IF chk = 1 THEN

IF amplt\$ = "C" AND cr\$ = "CR" THEN missed\$ = "CR" ELSE missed\$ = "TO"

freq\$ = "0"

IF missed\$ = "TO" THEN

```

    ReactionTime = 0
    amplt$ = "0"
END IF
WRITE #2, v, Viset, freq$, amplt$, ReactionTime, Totalex, missed$, TLC$
    Amp = VAL(Amp$(Table, t))
    freq = VAL(freq$(n))
    IF missed$ = "CR" THEN
        freq = 0
        Amp = 0
    END IF
    IF UCASE$(TLC$) = "T" THEN GOSUB Tgraph
    IF UCASE$(TLC$) = "L" THEN GOSUB Lgraph
    IF UCASE$(TLC$) = "C" THEN GOSUB Cgraph
    IF LEN(TLC$) < 1 THEN GOSUB Graph
    LPRINT USING fd$; v; "-"; Viset; freq; Amp; ReactionTime; missed$; t$
    amplt$ = ""
SelectNewVIAgain:
    pellet1 = TIMER + Viset
    LOCATE 17, 1: PRINT "New VI due TO error....."; Viset

END IF

NewVI3:
    pellet2 = TIMER
    IF INT(pellet2) >= INT(pellet1) + 5 THEN
        GOSUB SelectVI
        IF (TIMER + Viset) - INT(checkpull2) < 5 THEN GOTO NewVI3
        pellet1 = TIMER + Viset
        LOCATE 17, 1: PRINT "New VI due to error....."; Viset
    END IF
    LOCATE 12, 1: PRINT STRING$(50, " ")
    LOCATE 13, 1: PRINT USING "NEXT CONTRAST IN ### SECONDS "; Viset
    chk = 0
RETURN

CheckRight:
    checkpull2 = TIMER
HoldRight:
    firstpellet = 0
    SOUND 1200, .4
    IF Vlon = 3 AND stage = 3 THEN RleverON = RleverON + 1
    IF Vlon = 2 THEN RleverON = RleverON + 1
    LOCATE 22, 1: PRINT USING "Right lever ON....: #### "; RleverON
    lever2 = 0
DO
    TestPullRight = INP(770)
    LOCATE 9, 40: PRINT CHR$(4)
    LOCATE 10, 1: PRINT "Current Session Time.. "; TIMES$
    checkpull = TIMER
    IF Vlon > 0 THEN
        LOCATE 14, 1: PRINT "VI start number....: "; INT(checkpull), "Target number: "; INT(pellet1)
    END IF

```

```

    IF checkpull >= INT(show) THEN
        feedmanual = 0
        RETURN RFHR
    END IF
    LOOP UNTIL TestPullRight < 3
    LOCATE 9, 40: PRINT " "
    RETURN

```

```

    RETURN

```

```

CheckRightOFF:
    checkpull2 = TIMER

```

```

HoldRight2:
    SOUND 1600, .4
    DO
        rlever = INP(770)
        pellet3 = TIMER
        LOCATE 9, 40: PRINT CHR$(2)
        LOCATE 10, 1: PRINT "Current Session Time... "; TIME$
        checkpull = TIMER

        IF Vlon > 0 THEN
            LOCATE 14, 1: PRINT "VI start number... "; INT(checkpull), "Target number:"; INT(pellet1)
            END IF

        IF INT(pellet3) >= INT(pellet1) + 5 THEN
            GOSUB SelectVI
            missed$ = " "
            GOSUB WriteMissed
            pellet1 = TIMER + Vset
            LOCATE 12, 1: PRINT STRING$(50, " ")
            LOCATE 13, 1: PRINT USING "NEXT CONTRAST IN ### SECONDS "; Vset
            END IF
        LOOP WHILE rlever > 2

        RLeverOFF = RLeverOFF + 1
        LOCATE 23, 1: PRINT USING "Right lever OFF...: #### "; RLeverOFF
        lever2 = 0
        LOCATE 9, 40: PRINT " "
        RETURN

```

```

Setok:
    pp = pp + 1
    pulls = pulls + 1
    LOCATE 16, 1: PRINT "Lever was pulled less than 5 seconds before VI..: NO "
    LOCATE 17, 50: PRINT " "
    LOCATE 14, 1: PRINT "VI start number... "; INT(checkpull), "Target number:"; INT(pellet1)
    IF INT(pellet1) - INT(checkpull) < 5 THEN
        extendVI = extendVI + 1
        LOCATE 16, 51: PRINT "YES"
    END IF

```


NewVI:

```
GOSUB SelectVI
IF (TIMER + VIsset) - INT(checkpull) < 5 THEN GOTO NewVI
pellet1 = TIMER + VIsset
LOCATE 17, 1: PRINT "New VI due to error....."; VIsset
END IF
```

RETURN

ShowGrad:

```
IF UCASE$(trials$) = "Y" THEN ctial = ctial + 1 ' Count for catch trials
```

ShowGrad2:

```
DO
  levercheck = INP(768)
LOOP UNTIL levercheck < 5
missed$ = " "
IF st >= 2 AND VIon = 1 THEN
  VITM = VITM + 1
  LOCATE 16, 1: PRINT "Lever was pulled during 5 second hold time.....: NO"
  LOCATE 17, 50: PRINT " "
  LOCATE 14, 1: PRINT "VI start number...: "; INT(checkpull2), "Target number: "; INT(pellet1)

  LOCATE 16, 51: PRINT " NO"
  chck2 = TIMER + 5

  IF st > 1 AND st <> 2 THEN
    IF combo = 1 OR combo = 2 THEN n = INT(RND * freqs) + 1
    IF combo = 1 OR combo = 3 THEN t = INT(RND * av) + 1
  END IF

  IF TLC$ = "C" THEN
    t = INT(RND * av) + 1
  END IF
  LOCATE 9, 54: PRINT USING vvi$; v
  LOCATE 11, 54: PRINT USING ampl$; Amp$(Table, t)
  LOCATE 10, 54: PRINT USING fre$; freq$(n)
  LOCATE 12, 54: PRINT USING mode$; ActiveMode$

GOSUB Checkpull3
IF getgrad = 1 THEN GOTO getgrad
GOSUB WriteMissed
```

NewVI2:

```
GOSUB SelectVI
IF (TIMER + VIsset) - INT(checkpull2) < 5 THEN GOTO NewVI2
pellet1 = TIMER + VIsset
extendVI = extendVI + 1
LOCATE 17, 1: PRINT "New VI due to error....."; VIsset
RETURN
END IF
```

getgrad:

```

delay = TIMER
getgrad = 0
IF st >= 2 THEN LOCATE 16, 1: PRINT "Lever was pulled during 5 second hold time.....: YES"
show = TIMER + duration
show2 = TIMER
fd = 1

```

```

IF v > 4 AND UCASE$(trials$) = "Y" THEN
  IF Amp$(Table, t) = "C" AND trial = 0 THEN
    GOSUB ShowCatchTrial
    GOSUB CheckPull4
    IF missed$ = "C" THEN
      GOSUB WriteC
    END IF
    GOTO ByPass
  END IF
END IF

```

```

IF UCASE$(trials$) = "Y" THEN
  IF ctrial >= catch AND trial = 0 AND Vlon <> 0 AND v > 4 AND st >= 2 THEN
    GOSUB ShowCatchTrial
    trial = 0
    ctrial = 0
    GOSUB CheckPull4
    IF missed$ = "C" THEN
      GOSUB WriteC
    END IF
    GOTO ByPass
  END IF
END IF

```

```

PassCatch:
  IF Amp$(Table, t) = "C" THEN
    t = INT(RND * av) + 1
    IF Amp$(Table, t) = "C" THEN GOTO PassCatch
  END IF

```

```

IF ctrial = catch THEN trial = 0

```

```

IF TLC$ = "T" THEN
  IF set3 = 1 AND miss = 0 THEN
    last = 1
    t = t - 1
    IF t < 1 THEN t = 1
    IF Amp$(Table, t) = "C" THEN
      t = t - 1
      IF t < 1 THEN t = 1
    END IF
  END IF
END IF

```

```

IF set3 = 1 AND miss = 1 THEN
  last = 2

```

```

t = t + 1
IF t > av THEN t = av
IF Amp$(Table, t) = "C" THEN
    t = t + 1
    IF t > av THEN t = av
END IF
END IF
END IF
CheckC2:
IF TLC$ = "C" THEN
    VIon = 1
    t = INT(RND * av) + 1
    FOR l = 1 TO av
        IF Amp$(Table, t) = Amp$(Table, l) THEN
            IF total = (av * np) THEN GOTO endphase1
            IF count(l) >= np THEN GOTO CheckC2
            count(l) = count(l) + 1
        END IF
    NEXT l
END IF
OutputGrate:
miss = 0
PRINT #1, "FREQ=" + freq$(n)
PRINT #1, "AMP=" + Amp$(Table, t)
amplt$ = Amp$(Table, t)
IF VIon = 1 THEN VIon = 2 ELSE VIon = 3

ByPass:
LOCATE 9, 54: PRINT USING vvi$; v
LOCATE 11, 54: PRINT USING amplt$; amplt$
LOCATE 10, 54: PRINT USING fre$; freq$(n)
LOCATE 12, 54: PRINT USING mode$; ActiveMode$

CALL zeroCTM51(BSA%)
IF VIon = 1 THEN CALL readCTM51(ct(), BSA%)
IF stage = 3 THEN CALL readCTM51(ct(), BSA%)
IF st = 2 OR st = 3 THEN CALL readCTM51(ct(), BSA%)
begin = ct(1) / 1000
ReactionTime = 0
LOCATE 19, 1: PRINT USING "Reaction Time.....: ###.####"; ReactionTime
IF ctrial >= catch THEN ctrial = 0: trial = 0
RETURN

WriteMissed:
IF missed$ <> "C" THEN missed$ = "MVI"

ReactionTime = 0
freq$ = "0": amplt$ = "0"
WRITE #2, v, VIsset, freq$, amplt$, ReactionTime, Totalex, missed$, TLC$
    Amp = VAL(Amp$(Table, t))
    freq = VAL(freq$(n))

```

```

IF UCASE$(TLC$) = "T" THEN GOSUB Tgraph
IF UCASE$(TLC$) = "L" THEN GOSUB Lgraph
IF UCASE$(TLC$) = "C" THEN GOSUB Cgraph
IF LEN(TLC$) < 1 THEN GOSUB Graph
LPRINT USING fd$; v; "-"; Vset; freq; Amp; ReactionTime; missed$; t$

```

```
missed$ = ""
```

```
RETURN
```

```
WriteC:
```

```
missed$ = "C"
```

```
freq$ = "0"; amplt$ = "0"
```

```
WRITE #2, v, Vset, freq$, amplt$, ReactionTime, Totalex, missed$, TLC$
```

```
  Amp = VAL(Amp$(Table, t))
```

```
  freq = VAL(freq$(n))
```

```
  IF UCASE$(TLC$) = "T" THEN GOSUB Tgraph
```

```
  IF UCASE$(TLC$) = "L" THEN GOSUB Lgraph
```

```
  IF UCASE$(TLC$) = "C" THEN GOSUB Cgraph
```

```
  IF LEN(TLC$) < 1 THEN GOSUB Graph
```

```
  IF missed$ = "C" THEN
```

```
    freq = 0
```

```
    Amp = 0
```

```
  END IF
```

```
  LPRINT USING fd$; v; "-"; Vset; freq; Amp; ReactionTime; missed$; t$
```

```
  GOSUB SelectVI
```

```
  pellet1 = TIMER + Vset
```

```
RETURN
```

```
ShowCatchTrial:
```

```
  PRINT #1, "AMP=.005"
```

```
  Vion = 1
```

```
  trial = 1
```

```
  Cat = Cat + 1
```

```
  AlsCatch = 1
```

```
  ReactionTime = 0; missed$ = "C"
```

```
  amplt$ = Amp$(Table, t)
```

```
  IF amplt$ <> "C" THEN amplt$ = "C"
```

```
RETURN
```

```
Checkpull3:
```

```
  chck = 0
```

```
  ps = 0
```

```
DO
```

```
  levercheck = INP(768)
```

```
LOOP UNTIL levercheck < 5
```

```
  WHILE chck < chck2
```

```
    Checkpull3 = INP(768)
```

```

CheckPull4 = INP(770)
checkpull = TIMER
LOCATE 14, 1: PRINT "VI start number...: "; INT(checkpull), "Target number: "; INT(pellet1)
chk = TIMER
IF CheckPull4 > 2 THEN
    IF st >= 2 THEN RLeverOFF = RLeverOFF + 1
    getgrad = 1
    SOUND 2200, .4
    RETURN
END IF
IF Checkpull3 > 2 THEN
    chk = 1
    lh = 1
    delay2 = TIMER
    GOSUB CheckLeft2
    IF ps = 0 THEN GOTO BeginProcedure2
END IF
WEND
RETURN

```

TestPulls:

```

TestPullRight = INP(770)
TestPullLeft = INP(768)
LOCATE 9, 40: PRINT CHR$(2)
LOCATE 10, 1: PRINT "Current Session Time...: "; TIME$
checkpull = TIMER
IF Vlon > 0 THEN
    LOCATE 14, 1: PRINT "VI start number...: "; INT(checkpull), "Target number: "; INT(pellet1)
END IF
IF (checkpull >= (INT(pellet1) + duration)) THEN
    LOCATE 9, 40: PRINT " "
    RETURN 'BeginProcedure2
END IF
IF TestPullLeft > 2 THEN GOTO TestPulls
IF TestPullRight > 2 THEN GOTO TestPulls
LOCATE 9, 40: PRINT " "
RETURN

```

CheckPull4:

```

DO
    levercheck = INP(768)
LOOP UNTIL levercheck < 5
chk2 = TIMER + duration
chk = 0
ps = 0
WHILE chk < chk2
    Checkpull3 = INP(770)
    CheckPull4 = INP(768)
    chk = TIMER
    LOCATE 15, 72: PRINT "2 BAD"
    IF Checkpull3 > 2 THEN
        WHILE Checkpull3 > 2

```

```

    Checkpull3 = INP(770)
WEND
    RLeverOFF = RLeverOFF + 1
    LOCATE 23, 1: PRINT USING "Right lever OFF...: #### "; RLeverOFF
END IF

```

```

IF CheckPull4 > 2 THEN
    WHILE CheckPull4 > 2
        CheckPull4 = INP(768)
    WEND
    chk = 1
    cr$ = "CR"
    delay2 = delay
    GOSUB CheckLeft2
    IF ps = 0 THEN RETURN
END IF
WEND
RETURN

```

IncreaseTimeOut:

```

    Timeout = Timeout + 1
    LOCATE 21, 1: PRINT USING "Timeouts...: #### Current TO: ####"; LleverOff; Timeout
RETURN

```

DecreaseTimeOut:

```

    Timeout = Timeout - 1
    IF Timeout <= 0 THEN Timeout = 0
    LOCATE 21, 1: PRINT USING "Timeouts...: #### Current TO: ####"; LleverOff; Timeout
RETURN

```

IncreaseVI:

```

IF v = 0 THEN
    v = 4
    GOSUB InitVI
    RETURN
END IF

```

```

IF v = 4 THEN
    v = 6
    catch = 20
    GOSUB InitVI
    RETURN
END IF

```

```

IF v = 6 THEN
    v = 8
    catch = 12
    GOSUB InitVI
    RETURN
END IF

```

IF v = 8 THEN
v = 10
catch = 11
GOSUB InitVI
RETURN
END IF

IF v = 10 THEN
v = 12
catch = 10
GOSUB InitVI
RETURN
END IF

IF v = 12 THEN
v = 16
catch = 8
GOSUB InitVI
RETURN
END IF

IF v = 16 THEN
v = 20
catch = 5
GOSUB InitVI
RETURN
END IF

IF v = 20 THEN
v = 23
catch = 5
GOSUB InitVI
RETURN
END IF

IF v = 23 THEN
v = 24
catch = 5
GOSUB InitVI
RETURN
END IF

IF v = 24 THEN
v = 26
catch = 4
GOSUB InitVI
RETURN
END IF

IF v = 26 THEN

```
v = 30
catch = 4
GOSUB InitVI
RETURN
END IF
```

```
IF v = 30 THEN
  v = 36
  catch = 3
  GOSUB InitVI
  RETURN
END IF
```

```
IF v = 36 THEN
  v = 55
  catch = 1
  GOSUB InitVI
  RETURN
END IF
```

```
IF v = 55 THEN
  v = 144
  catch = 1
  GOSUB InitVI
  RETURN
END IF
```

```
IF v = 144 THEN
  v = 144
  catch = 1
  GOSUB InitVI
  RETURN
END IF
```

```
DecreaseVI:
IF v = 0 THEN
  v = 0
  GOSUB InitVI
  RETURN
END IF
```

```
IF v = 4 THEN
  v = 4
  GOSUB InitVI
  RETURN
END IF
```

```
IF v = 6 THEN
  v = 4
  catch = 30
  GOSUB InitVI
```


RETURN
END IF

IF v = 8 THEN
v = 6
catch = 20
GOSUB InitVI
RETURN
END IF

IF v = 10 THEN
v = 8
catch = 12
GOSUB InitVI
RETURN
END IF

IF v = 12 THEN
v = 10
catch = 11
GOSUB InitVI
RETURN
END IF

IF v = 16 THEN
v = 12
catch = 10
GOSUB InitVI
RETURN
END IF

IF v = 20 THEN
v = 16
catch = 8
GOSUB InitVI
RETURN
END IF

IF v = 23 THEN
v = 20
catch = 5
GOSUB InitVI
RETURN
END IF

IF v = 24 THEN
v = 23
catch = 5
GOSUB InitVI
RETURN
END IF

```

IF v = 26 THEN
  v = 24
  catch = 5
  GOSUB InitVI
  RETURN
END IF

```

```

IF v = 30 THEN
  v = 26
  catch = 5
  GOSUB InitVI
  RETURN
END IF

```

```

IF v = 36 THEN
  v = 30
  catch = 4
  GOSUB InitVI
  RETURN
END IF

```

```

IF v = 55 THEN
  v = 36
  catch = 3
  GOSUB InitVI
  RETURN
END IF

```

```

IF v = 144 THEN
  v = 55
  catch = 3
  GOSUB InitVI
  RETURN
END IF

```

```

InitVI:
  GOSUB SelectVI
  PRINT #1, "AMP=0"
  pellet1 = TIMER + V1set
  fcr1 = TIMER + FCR
  icr1 = TIMER + ICR
  LOCATE 9, 54: PRINT USING w1$; v
  Changed = Changed + 1
  V1on = 1
  CALL zeroCTM51(BSA%)
  fd = 0
  IF st >= 1 THEN phase = 2
  IF st >= 2 THEN phase = 4
  tcurrenttime = 0: tavcount = 0

```

```

tcurrent = 0; tctime = 0
tavetimer = TIMER
taveragetime = 0
setfirstpelletonvichange = setfirstpelletonvichange + 1
IF setfirstpelletonvichange = 1 THEN
    first = TIMER
END IF
first2 = TIMER
averagepellets = 0
RETURN

```

SelectVI:

```

var = INT(RND * 30) + 1
Vset = VI(v, var)
IF AlsCatch = 1 AND Vset > 15 THEN GOTO SelectVI
AlsCatch = 0
RETURN

```

FreqChange:

```

IF combo = 1 OR combo = 2 THEN n = INT(RND * freqs) + 1
IF st > 1 THEN PRINT #1, "AMP=0"
IF testphase = 1 AND n = 6 THEN GOTO FreqChange
PRINT #1, "FREQ=" + freq$(n)
fcr1 = TIMER + FCR
LOCATE 10, 54: PRINT USING fre$; freq$(n)
RETURN

```

IntensityChange:

```

IF combo = 1 OR combo = 3 THEN t = INT(RND * av) + 1
IF Amp$(Table, t) = "C" THEN GOTO IntensityChange
IF ss = 1 OR phase = 3 OR st = 2 THEN GOTO Skint
PRINT #1, "AMP=" + Amp$(Table, t)

```

Skint:

```

icr1 = TIMER + ICR
LOCATE 11, 54: PRINT USING ampl$; Amp$(Table, t)
RETURN

```

GetReinforcement:

```

GOSUB GetReactionTime
mc = 0
IF fd = 1 THEN
    GOSUB FeedPellet
    GOSUB GetAverage
    LOCATE 23, 44: PRINT USING "Number Pellets Received.....: ###"; count + Fed
    pullis = pullis + 1
    contrastON = contrastON + 1
    IF SleepTime > 0 OR Vlon <> 0 THEN
        PRINT #1, "AMP=0"
        PRINT #1, "FREQ=1100 ME ON"
    END IF
    total = total + Vset
    avetimer = TIMER

```

tavetimer = TIMER

missed\$ = " "

WRITE #2, v, Vset, freq\$(n), Amp\$(Table, t), ReactionTime, Totalex, missed\$, UCASE\$(TLC\$)

Amp = VAL(Amp\$(Table, t))

freq = VAL(freq\$(n))

IF UCASE\$(TLC\$) = "T" THEN GOSUB Tgraph

IF UCASE\$(TLC\$) = "L" THEN GOSUB Lgraph

IF UCASE\$(TLC\$) = "C" THEN GOSUB Cgraph

IF LEN(TLC\$) < 1 THEN GOSUB Graph

LPRINT USING fd\$, v, " -"; Vset; freq; Amp; ReactionTime; missed\$, t\$

GOSUB TestPulls

GOSUB SelectVI

pellet1 = TIMER + Vset

delay = TIMER + SleepTime

IF phase1 = 1 THEN

LOCATE 11, 2

PRINT "NEXT PELLET IN "; Vset; " SECONDS "

END IF

END IF

fd = 1

LeverHold:

IF Vion = 3 THEN Vion = 1

RETURN

GetReactionTime:

IF Vion > 0 THEN CALL readCTM51(ct(), BSA%)

IF Vion = 0 THEN pulls = pulls + 1

endpull = ct(1) / 1000

IF Vion > 0 THEN

ReactionTime = endpull - begin

LOCATE 19, 1

PRINT USING "Reaction Time.....: ###.####"; ReactionTime

ELSE ReactionTime = 0

END IF

RETURN

GetAverage:

IF manualfeed = 0 THEN

LleverON = LleverON + 1

count = count + 1

averagepellets = averagepellets + 1

END IF

IF v > 0 THEN

IF averagepellets > 0 THEN

averagetime = (pelletavetimer - first2) / averagepellets

END IF

averagetimer = TIMER

taveragetime = (averagetimer - first) / count

```

        LOCATE 21, 44: PRINT USING "Session VI.....: ###.##"; taveragetime
        LOCATE 22, 44: PRINT USING "Interval for current VI.....: ###.##"; averagetime
    END IF
    RETURN

FeedPellet:
    OUT 777, 4
    FOR x = 1 TO 3500: NEXT x
    OUT 777, 0
    IF Vion = 1 OR Vion = 2 THEN
        LOCATE 11, 54: PRINT USING amp$; "0.0"
        PRINT #1, "AMP=0"
    END IF
    delay = TIMER
    RETURN

IncreaseAmplitude:
    t = t + 1
    IF Amp$(Table, t) = "C" THEN t = t + 1
    IF t > av THEN t = av
    LOCATE 11, 54: PRINT USING amp$; Amp$(Table, t)
    IF st = 1 AND Vion = 0 THEN PRINT #1, "AMP=" + Amp$(Table, t)
    RETURN

DecreaseAmplitude:
    t = t - 1
    IF Amp$(Table, t) = "C" THEN t = t - 1
    IF t = 0 THEN t = 1
    LOCATE 11, 54: PRINT USING amp$; Amp$(Table, t)
    IF st = 1 AND Vion = 0 THEN PRINT #1, "AMP=" + Amp$(Table, t)
    RETURN

IncreaseFrequency:
    n = n + 1
    IF n > freqs THEN n = freqs
    LOCATE 10, 54: PRINT USING fre$; freq$(n)
    PRINT #1, "FREQ=" + freq$(n)
    RETURN

DecreaseFrequency:
    n = n - 1
    IF n = 0 THEN n = 1
    LOCATE 10, 54: PRINT USING fre$; freq$(n)
    PRINT #1, "FREQ=" + freq$(n)
    RETURN

Feed:
    IF INT(pellet1) - INT(checkpull) < 5 THEN RETURN
    lever2 = 0
    manualfeed = 1
    LOCATE 19, 47: PRINT " Press <SPACE BAR> to feed "
    x$ = ""
    ReactionTime = 0
    LOCATE 19, 1: PRINT USING "Reaction Time.....: ###.####"; ReactionTime

```

```

CALL zeroCTM51(BSA%)
CALL readCTM51(ct(), BSA%)
begin = ct(1) / 1000
PRINT #1, "FREQ=" + freq$(n)
PRINT #1, "AMP=20"
ManualGrates = ManualGrates + 1

DO UNTIL (lever2 > 2) OR x$ = " "
  x$ = INKEY$
  lever2 = INP(768)
LOOP

GOSUB GetAverage

manualfeed = 0

IF x$ <> " " THEN
  Fed = Fed - 1
  count = count + 1
  LleverON = LleverON + 1
  GOSUB GetReactionTime
END IF

Feed2:
x$ = ""
LOCATE 19, 47: PRINT SPACE$(27)

OUT 777, 4
FOR x = 1 TO 3500: NEXT x
OUT 777, 0

IF Vlon <> 0 THEN
  PRINT #1, "AMP=0"
END IF
Fed = Fed + 1
IF phase = 0 THEN
  LOCATE 13, 28
  PRINT USING "###"; Fed
ELSE
  LOCATE 23, 44: PRINT USING "Number Pellets Received.....: ###"; count + Fed
END IF
GOSUB SelectVI

LOCATE 12, 1: PRINT STRING$(50, " ")
LOCATE 13, 1: PRINT USING "NEXT CONTRAST IN ### SECONDS "; V1set
pellet1 = TIMER + V1set
avetimer = TIMER
tavetimer = TIMER
Vlon = 2
fedmanual = 1
delay = TIMER + SleepTime

```

RETURN

EntryScreen:

```
end1 = 0
averagetime = 0
current = 0
avcount = 0: ctime = 0
currenttime = 0: avetimer = 0
taveragetime = 0
tcurrent = 0
tavcount = 0: tctime = 0
tcurrenttime = 0: tavetimer = 0

count = 0
start1 = 0
total = 0
Vlon = 0
set3 = 0
trials$ = ""
na$ = "": y1$ = "": y2$ = ""
CLS
IF a$ = "0" OR a$ = "1" OR a$ = "2" OR a$ = "3" THEN av = 18
IF a$ = "4" THEN av = 42
IF stage$ <> "Stage 3" THEN
    LOCATE 2, 27: PRINT stage$, " Training Procedure"
END IF
IF stage$ = "Stage 3" THEN
    set3 = 1
    a$ = "4"
    LOCATE 2, 20: PRINT stage$, " - Contrast Sensitivity Procedure "
END IF
LOCATE 5, 20: PRINT "Enter file name.....: "
IF VAL(a$) = 4 THEN na$ = "N/A"
LOCATE 6, 20: PRINT "Do you want random frequencies (Y/N): "
IF na$ = "N/A" THEN
    LOCATE 6, 20: PRINT "(T)racking, (L)imits, (C)onstant....: "
    y2$ = na$
END IF
LOCATE 7, 20: PRINT "Do you want random intensities (Y/N): "; na$

LOCATE 8, 20: PRINT "Session length in minutes.....: "

IF stage$ = "Stage 1" OR stage$ = "Stage 0" THEN
    na$ = "N/A"
ELSE na$ = ""
END IF
IF phase = 0 THEN
    LOCATE 9, 20: PRINT "VI Rate.....: "
    GOTO Skip9
END IF
LOCATE 9, 20: PRINT "Sleep Time in seconds.....: "; na$
```

Skip9:

```
LOCATE 10, 20: PRINT "Time-Out in seconds.....: "; na$
LOCATE 11, 20: PRINT "Duration of Contrast.....: "; na$
LOCATE 12, 20: PRINT "Do you want catch trials (Y/N).....: "; na$
```

SkipEntry:

```
LOCATE 5, 58: INPUT ">", file$
IF LEN(file$) < 1 THEN GOTO SkipEntry
```

FileCheck:

```
IF DIR$(file$) = UCASE$(file$) THEN ' Check to see if file already
                                     ' exists in directory
```

BEEP

```
LOCATE 22, 10
```

```
PRINT "File already exists - (W)rite Over or (N)ew Name?"
```

DO

```
qq$ = INKEY$
```

```
LOOP UNTIL UCASE$(qq$) = "W" OR UCASE$(qq$) = "N"
```

```
LOCATE 22, 10: PRINT SPACE$(50)
```

```
IF UCASE$(qq$) = "N" THEN
```

```
LOCATE 5, 59: PRINT SPACE$(15)
```

```
GOTO SkipEntry
```

```
END IF
```

```
END IF
```

```
LOCATE 5, 58: PRINT " "
```

TLC:

```
LOCATE 6, 58: INPUT ">", y1$
```

```
y1$ = UCASE$(y1$)
```

```
IF stage$ = "Stage 3" THEN
```

```
IF (y1$ = "T" OR y1$ = "L" OR y1$ = "C") THEN
```

```
GOTO TLC2
```

```
ELSE GOTO TLC
```

```
END IF
```

```
END IF
```

TLC2:

```
LOCATE 6, 58: PRINT " "
```

```
IF VAL(a$) = 4 THEN GOTO SkEntry
```

```
LOCATE 7, 58: INPUT ">", y2$
```

```
LOCATE 7, 58: PRINT " "
```

SkEntry:

```
IF y1$ = "C" THEN LOCATE 8, 58: PRINT "N/A": GOTO SkipSession
```

```
LOCATE 8, 58: INPUT ">", SessionLength
```

```
LOCATE 8, 58: PRINT " "
```

SkipSession:

```
IF stage$ = "Stage 1" OR stage$ = "Stage 0" THEN GOTO SkipEntry2
```

inVI:

```
IF phase = 0 THEN
```

```
LOCATE 9, 58: INPUT ">", v
```

```
LOCATE 9, 58: PRINT " "
```

```
END IF
```

```
LOCATE 9, 58: INPUT ">", SleepTime
```

```
LOCATE 9, 58: PRINT " "
```


SkipSleep:

```
LOCATE 10, 58: INPUT ">", Timeout
LOCATE 10, 58: PRINT " "
LOCATE 11, 58: INPUT ">", duration
LOCATE 11, 58: PRINT " "
LOCATE 12, 58: INPUT ">", trials$
LOCATE 12, 58: PRINT " "
```

SkipEntry2:

```
IF phase = 0 THEN
    LOCATE 9, 58: INPUT ">", v
    LOCATE 9, 58: PRINT " "
END IF
```

length = SessionLength * 60

start = TIMER

IF phase <> 0 THEN

OPEN file\$ FOR OUTPUT AS #2

WRITE #2, y1\$, y2\$, SessionLength, SleepTime, Timeout, duration

END IF

COMBO1:

IF UCASE\$(y1\$) = "Y" AND UCASE\$(y2\$) = "Y" THEN

LOCATE 15, 17: PRINT "You selected random frequencies and intensities"

LOCATE 17, 17: INPUT "Frequency change rate in minutes.....: ", FCR

LOCATE 18, 17: INPUT "Intensity change rate in minutes.....: ", ICR

FCR = FCR * 60

ICR = ICR * 60

n = INT(RND * freqs) + 1

C1:

t = INT(RND * av) + 1

IF Amp\$(Table, t) = "C" THEN GOTO C1

IF phase <> 0 THEN

WRITE #2, FCR / 60, ICR / 60, freq\$(n), Amp\$(Table, t)

END IF

combo = 1

GOSUB TestKey2

END IF

COMBO2:

IF UCASE\$(y1\$) = "Y" AND UCASE\$(y2\$) <> "Y" THEN

LOCATE 15, 17: PRINT "You selected random frequencies and constant intensities"

LOCATE 17, 17: INPUT "Frequency change rate in minutes.....: ", FCR

n = INT(RND * freqs) + 1

FCR = FCR * 60

combo = 2

SetVolts:

LOCATE 18, 17: INPUT "Enter valid intensity: ", volts\$

ICR = 0

```

    FOR t = 1 TO av
      IF Amp$(Table, t) = volts$ THEN GOTO GoAhead2
    NEXT t
    BEEP
    GOTO SetVolts
GoAhead2:
    IF phase <> 0 THEN
      WRITE #2, FCR / 60, ICR, freq$(n), Amp$(Table, t)
    END IF
    GOSUB TestKey2
  END IF

COMBO3:

  IF UCASE$(y1$) <> "Y" AND UCASE$(y2$) = "Y" THEN
    LOCATE 15, 17: PRINT "You selected a constant frequency and random intensities"
  SetFreq1:
    LOCATE 17, 17: INPUT "Enter a valid frequency.....: ", frq$
    FOR n = 1 TO freqs
      IF frq$ = freq$(n) THEN GOTO goahead3
    NEXT n
    BEEP
    GOTO SetFreq1
  goahead3:
    LOCATE 18, 17: INPUT "Intensity change rate in minutes.....: ", ICR
    ICR = ICR * 60
    FCR = 0
  C3:
    t = INT(RND * av) + 1
    IF Amp$(Table, t) = "C" THEN GOTO C3
    combo = 3
    IF phase <> 0 THEN
      WRITE #2, FCR, ICR / 60, freq$(n), Amp$(Table, t)
    END IF
    GOSUB TestKey2
  END IF

COMBO4:
  IF UCASE$(y1$) = "C" THEN
    LOCATE 15, 17: PRINT "You selected Constant Stimuli....."
    GOTO GoAhead4
  END IF
  IF UCASE$(y1$) <> "Y" AND UCASE$(y2$) <> "Y" THEN
    LOCATE 15, 17: PRINT "You selected constant frequency and intensity"
  SetFreq2:
    LOCATE 17, 17: INPUT "Enter valid frequency.....: ", frq$

    FOR n = 1 TO freqs
      IF frq$ = freq$(n) THEN GOTO SetVolts2
    NEXT n
    BEEP
    GOTO SetFreq2

```

SetVolts2:

```
LOCATE 18, 17: INPUT "Enter valid intensity.....: ", volts$
FOR t = 1 TO av
  IF volts$ = Amp$(Table, t) THEN GOTO GoAhead4
NEXT t
BEEP
GOTO SetVolts2
```

GoAhead4:

```
combo = 4
ICR = 0
FCR = 0
TLC$ = y1$
IF TLC$ = "C" THEN freq$ = "N/A": Amp$ = "N/A"
freq$ = freq$(n): Amp$ = Amp$(Table, t)
IF phase <> 0 THEN
  WRITE #2, FCR, ICR, freq$, Amp$
END IF
GOSUB TestKey2
END IF

RETURN
```

TestKey2:

```
a$ = ""
COLOR 0, 7
LOCATE 22, 24: PRINT "Information Correct? (Y/N)"
SOUND 470, .4
COLOR 7, 0
a$ = INPUT$(1)
IF UCASE$(a$) <> "Y" THEN
  CLOSE #2
  GOTO EntryScreen
END IF
LOCATE 22, 1: PRINT SPACE$(70)
RETURN
```

Tgraph:

```
ta$ = LTRIM$(STR$(Amp))
FOR tcheck = 1 TO amps
  IF ta$ = Amp$(Table, tcheck) THEN
    t$ = SPACE$(tcheck) + "T"
    RETURN
  END IF
NEXT tcheck
RETURN
```

Lgraph:

```
ta$ = LTRIM$(STR$(Amp))
FOR tcheck = 1 TO amps
```

```

IF ta$ = Amp$(Table, tcheck) THEN
  t$ = SPACE$(tcheck) + "L"
  RETURN
END IF
NEXT tcheck
RETURN

```

Cgraph:

```

ta$ = LTRIM$(STR$(Amp))
FOR tcheck = 1 TO amps
  IF ta$ = Amp$(Table, tcheck) THEN
    t$ = SPACE$(tcheck) + "C"
    RETURN
  END IF
NEXT tcheck
RETURN

```

Graph:

```

ta$ = LTRIM$(STR$(Amp))
FOR tcheck = 1 TO amps
  IF ta$ = Amp$(Table, tcheck) THEN
    t$ = SPACE$(tcheck) + "*"
    RETURN
  END IF
NEXT tcheck
RETURN

```

endphase1:

```

CLS
FinalSessionLengthSeconds = TIMER
FSL = FinalSessionLengthSeconds - SessionLengthSeconds
TIMER OFF
PRINT #1, "ME OFF"
PRINT #1, "HIDE"
PRINT #1, "FREQ=0 AMP=0"
OUT 777, 0
KEY OFF
IF phase = 0 THEN
  LPRINT "Subject Number.....: "; file$
  LPRINT "Date: "; DATE$
  LPRINT "Time: "; TIME$
  LPRINT
  LPRINT USING "Number of pellets received.....: ####"; Fed
  LPRINT USING "Session Length (minutes).....: ####"; SessionLength
  LPRINT USING "Average VI.....: ###.###"; length / Fed
  LPRINT CHR$(12)
  GOTO mainmenu

```

OUT 779, 136

END IF

IF TLC\$ = "C" THEN GOTO EndPhase2

INPUT "Extend Session Time (Y/N) "; y\$

IF UCASE\$(y\$) = "Y" THEN

INPUT "Enter Extended Session Time (Minutes): ", exsessiontime

Totalex = Totalex + exsessiontime

length = exsessiontime * 60

fcr1 = TIMER + FCR

icr1 = TIMER + ICR

current = 0

avetimer = 0

ctime = 0

tctime = 0

tavetimer = 0

PRINT #1, "SHOW ME ON": KEY ON

KEY(10) OFF

ON KEY(10) GOSUB endphase1

KEY(10) ON

PRINT #1, "AMP=0"

IF phase = 1 OR phase = 2 THEN GOSUB Bp

IF phase = 3 OR phase = 4 THEN GOSUB Bp2

END IF

CLS

EndPhase2:

SOUND 470, .4

COLOR 0, 7

LOCATE 10, 21: PRINT " Printing in Progress - Please Standby "

COLOR 7, 0

CLOSE #2

OPEN file\$ FOR INPUT AS #2

OPEN "XXXXXX.DAT" FOR OUTPUT AS #3

LPRINT

WRITE #3, file\$, DATE\$, TIME\$

LPRINT

LPRINT

missed = 0

trans = 0

MVI = 0

ave = 0

tt = 0

t\$ = " C "

INPUT #2, y1\$, y2\$, SessionLength, SleepTime

INPUT #2, Timeout, duration, FCR, ICR, freq\$, Amp\$

```

INPUT #2, st, av
'freq$ = freq$
IF y1$ = "TRACKING" THEN y1$ = "T"
IF y1$ = "LIMITS" THEN y1$ = "L"
IF y1$ = "CONSTANT STIMULI" THEN y1$ = "C"
TLC$ = UCASE$(y1$)
IF UCASE$(y1$) = "T" THEN y1$ = "TRACKING"
IF UCASE$(y1$) = "L" THEN y1$ = "LIMITS"
IF UCASE$(y1$) = "C" THEN y1$ = "CONSTANT STIMULI"
  LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
  LPRINT "INITIAL SETTINGS:"
  LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
  LPRINT
  LPRINT "Random Frequencies.....: "; UCASE$(y1$)
  LPRINT "Random Amplitudes.....: "; UCASE$(y2$)
  LPRINT
  LPRINT "Sessionlength (Min).....: "; SessionLength
  LPRINT
  LPRINT "Frequency Rate Change (Min): "; FCR
  LPRINT "Amplitude Rate Change (Min): "; ICR
  LPRINT
  LPRINT "Initial Frequency.....: "; freq$
  LPRINT "Initial Amplitude.....: "; Amp$
  LPRINT
  LPRINT "Sleep Time (Seconds).....: "; SleepTime
  LPRINT "Time Out (Seconds).....: "; Timeout
  LPRINT
  LPRINT "Duration of Grating (Sec)...: "; duration
  LPRINT "Catch Trials (Y/N).....: "; UCASE$(trials$)
  LPRINT

WRITE #3, stage$, y1$, y2$, SessionLength, FCR, ICR, freq$, Amp$
WRITE #3, SleepTime, Timeout, duration, UCASE$(trials$)
WRITE #3, st, av
  x = 18
  ave = 0
  average = 0
  WHILE NOT EOF(2)
    INPUT #2, v, V1set, freq$, Amp$, ReactionTime, Totalex, missed$, TLC$
    IF missed$ = "XXX" THEN GOTO CheckPrevious
    x = x + 1
    IF missed$ = " " THEN average = average + 1
    freq = VAL(freq$); Amp = VAL(Amp$)
    Tamp = Amp
    Lamp = Amp
    tt = tt + 1
    Tamp$(tt) = LTRIM$(STR$(Tamp))
    Lamp$(tt) = Lamp$(tt)

  IF st = 3 THEN
    IF UCASE$(TLC$) = "T" OR UCASE$(TLC$) = "C" THEN
      FOR loadtt = 1 TO av

```

```

    IF Tamp$(tt) = Amp$(Table, loadtt) THEN
        Tamp(loadtt) = Tamp(loadtt) + 1
        IF (missed$ = " ") OR (missed$ = "EPC") THEN
            TampC(loadtt) = TampC(loadtt) + 1
            rt(loadtt) = rt(loadtt) + ReactionTime
        END IF
    END IF
NEXT loadtt
END IF

IF UCASE$(TLC$) = "L" THEN
    FOR loadll = 1 TO av
        IF Lamp$(tt) = Amp$(Table, loadll) THEN
            Lamp(loadll) = Lamp(loadll) + 1
            IF (missed$ = " ") OR (missed$ = "EPC") THEN
                LampC(loadll) = LampC(loadll) + 1
                rt(loadll) = rt(loadll) + ReactionTime
            END IF
        END IF
    NEXT loadll
END IF
END IF

IF missed$ = "MC" THEN
    missed = missed + 1
END IF
IF missed$ = "MVI" THEN MVI = MVI + 1
IF missed$ = "CR" THEN CatchTrials = CatchTrials + 1
IF missed$ = " " THEN ave = ave + VIsset
t$ = ""
IF UCASE$(TLC$) = "T" THEN GOSUB Tgraph
IF UCASE$(TLC$) = "L" THEN GOSUB Lgraph
IF UCASE$(TLC$) = "C" THEN GOSUB Cgraph
IF LEN(TLC$) < 1 THEN GOSUB Graph
WRITE #3, v, VIsset, freq, Amp, ReactionTime, missed$
WEND

CheckPrevious:
vv = 0: vviset = 0: ffreq = 0: aamp = 0: rreactiontime = 0
mi$ = "XXX"
WRITE #3, vv, vviset, ffreq, aamp, rreactiontime, mi$
aveVI = ave / average
IF previous = 1 THEN
    INPUT #2, trials, count, Fed, ContrastOFF, RLeverOFF, RLeverON
    INPUT #2, LLeverOff, LLeverON, PullSleep, Cat, CatchTrials, Changed
    INPUT #2, Timeout
END IF
LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
LPRINT "FINAL RESULTS:"
LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
LPRINT
LPRINT USING "Pellets Received as a Result of Work.....: ####"; count
LPRINT USING "Manual Pellets Received.....: ####"; Fed
LPRINT USING "Manual Grates Presented.....: ####"; ManualGrates

```

```

LPRINT USING "Total Pellets Received.....: ####"; count + Fed
IF st = 1 THEN
  LPRINT USING "Total Responses with Contrast ON.....: ####"; count
  LPRINT USING "Total Responses with Contrast OFF.....: ####"; ContrastOFF
  LPRINT USING "Total Responses during session.....: ####"; count + ContrastOFF
END IF

```

```

IF st >= 2 THEN
  LPRINT USING "Right Lever Responses with Contrast OFF.....: ####"; RLeverOFF
  LPRINT USING "Right Lever Responses with Contrast ON (Errors).....: ####"; RLeverON
  LPRINT USING "Left Lever Responses with Contrast OFF (Errors/TO's)..: ####"; LLeverOff
  LPRINT USING "Left Lever Responses with Contrast ON.....: ####"; LLeverON
  LPRINT USING "Left Lever Responses During Sleep Time.....: ####"; PullSleep
  LPRINT USING "Total Catch Trials Presented.....: ####"; Cat
  LPRINT USING "Total Catch Trials Responded To.....: ####"; CatchTrials
  LPRINT USING "Missed Contrasts.....: ####"; missed
  LPRINT USING "Number of VI scheduled trials missed by subject.....: ####"; MVI
END IF
  LPRINT USING "Average Time between pellets.....: ####.##"; taveragetime
  LPRINT USING "VI changes during this session.....: ####"; Changed
  LPRINT USING "Ending TIMEOUT length.....: ####"; Timeout
  LPRINT USING "Session Time Extended (Minutes).....: ####"; Totalex
  LPRINT
  LPRINT USING "Session Length (min)....: #####.##"; FSL / 60
  LPRINT USING "Right Lever Responses/sec...: #####.##"; RLeverOFF / FSL
  LPRINT

```

```

WRITE #3, trials, count, Fed, ContrastOFF, RLeverOFF, RLeverON
WRITE #3, LLeverOff, LLeverON, PullSleep, Cat, CatchTrials, missed
WRITE #3, MVI, taveragetime, Changed, Timeout, Totalex

```

```

IF st = 3 THEN

```

```

ov1$ = " \      \ \      \ \      \      \      \      \"
ov2$ = "
ov3$ = "      ###      ##.###      ###      ###      ###.##      ###.## "

```

```

IF y1$ <> "LIMITS" THEN
  LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
  LPRINT "Ordered Voltages Tested: "; y1$
  LPRINT
  LPRINT "Frequency Tested: "; frq$; " Hz"
  LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
  yy$ = LEFT$(y1$, 1)
  LPRINT

```

```

LPRINT USING ov1$, "Contrast Sensitivity"; "Voltage"; "No. Presented"; "Correct Responses"; "Mean RT"

```

```

LPRINT USING ov2$, "No."; "% Correct"
LPRINT STRING$(100, "-")
LPRINT

```



```

FOR pt = 1 TO av
  IF Tamp(pt) > 0 THEN
    IF TampC(pt) > 0 THEN
      Ampc = VAL(Amp$(Table, pt))
      GOSUB GetContrast
      LPRINT USING ov3$; Ampb; VAL(Amp$(Table, pt)); Tamp(pt); TampC(pt); (TampC(pt) /
        Tamp(pt)) * 100; rt(pt) / TampC(pt)
    ELSE
      Ampc = VAL(Amp$(Table, pt))
      GOSUB GetContrast
      LPRINT USING ov3$; Ampb; VAL(Amp$(Table, pt)); Tamp(pt); TampC(pt); (TampC(pt) /
        Tamp(pt)) * 100; 0
    END IF

    IF TampC(pt) > 0 THEN
      WRITE #3, yy$, Ampb, VAL(Amp$(Table, pt)), Tamp(pt), TampC(pt), (TampC(pt) /
        Tamp(pt)) * 100, rt(pt) / TampC(pt)
    ELSE WRITE #3, yy$, Ampb, VAL(Amp$(Table, pt)), Tamp(pt), TampC(pt), (TampC(pt) /
      Tamp(pt)) * 100, 0
    END IF
  END IF
NEXT pt

```

LPRINT : LPRINT

ELSE

'IF y1\$ = "CONSTANT STIMULI" THEN GOTO skiplimits

```

LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
LPRINT "Ordered Voltages Tested: LIMITS "
LPRINT
LPRINT "Frequency Tested: "; frq$; " Hz"
LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
LPRINT
LPRINT USING ov1$; "Contrast Sensitivity"; "Voltage"; "No. Presented"; "Correct Responses";
  "Mean RT"
LPRINT USING ov2$; "No."; "% Correct"
LPRINT STRING$(100, "-")
LPRINT
Limit$ = "L"
FOR pt = 1 TO av
  IF Lamp(pt) > 0 THEN
    IF LampC(pt) > 0 THEN
      Ampc = VAL(Amp$(Table, pt))
      GOSUB GetContrast
      LPRINT USING ov3$; Ampb; VAL(Amp$(Table, pt)); Lamp(pt); LampC(pt); (LampC(pt) /
        Lamp(pt)) * 100; rt(pt) / LampC(pt)
    ELSE
      Ampc = VAL(Amp$(Table, pt))
      GOSUB GetContrast
      LPRINT USING ov3$; Ampb; VAL(Amp$(Table, pt)); Lamp(pt); LampC(pt); (LampC(pt) /

```

```

        Lamp(pt)) * 100; 0
    END IF

    IF LampC(pt) > 0 THEN
        WRITE #3, Limit$, Ampb, VAL(Amp$(Table, pt)), Lamp(pt), LampC(pt), (LampC(pt) /
            Lamp(pt)) * 100, rt(pt) / LampC(pt)
        ELSE WRITE #3, Limit$, Ampb, VAL(Amp$(Table, pt)), Lamp(pt), LampC(pt),
            (LampC(pt) / Lamp(pt)) * 100, 0
    END IF
END IF
NEXT pt

END IF
END IF
skiplimits:
    LPRINT
    LPRINT
    LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
    LPRINT "Label Codes:"
    LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
    LPRINT
    LPRINT " VI - Fixed Interval      Freq - Frequency (Hz)"
    LPRINT " Amp - Amplitude (Volts)   RT - Reaction Time (sec)"
    LPRINT
    LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
    LPRINT "Symbol Codes: (SC)"
    LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
    LPRINT
    LPRINT " TO = Timeout      MC = Missed Contrast   MVI = Missed scheduled VI"
    LPRINT " C = Catch Trial (No response)   CR = Catch Trial (Response)"
    LPRINT " EPC = Experimenter Presented Contrast"
    LPRINT
    LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
    LPRINT "Procedure Codes:"
    LPRINT CHR$(27); CHR$(72); CHR$(27); CHR$(87);
    LPRINT
    LPRINT " T = Tracking    L = Limits    C = Contrast    * = Other"

    LPRINT CHR$(12)

    IF previous = 0 THEN
        CLOSE #2
        OPEN file$ FOR APPEND AS #2
        WRITE #2, trials, count, Fed, ContrastOFF, RLeverOFF, RLeverON
        WRITE #2, LLeverOff, LLeverON, Cat, CatchTrials, Changed
        WRITE #2, Timeout
    END IF

    CLOSE #2
    CLOSE #3
    KILL file$
    NAME "XXXXXX.DAT" AS file$

```

KEY OFF

```
IF y1$ = "CONSTANT STIMULI" THEN
  FOR loadamp = 1 TO 18      ' Re-Load Amplitudes
    READ Amp$(Table, loadamp)
  NEXT loadamp
END IF
CLS
END
```

GetContrast:

```
FOR match = 1 TO contrasts
  IF Ampc = Voltage(match) THEN
    Ampb = Sensitivity(match)
  END IF
NEXT match
RETURN
```

errcheck:

SOUND 470, .4

errcheck2:

```
SELECT CASE ERR
  CASE 6
    FOR x = 1 TO 2
      SOUNDOn = SOUNDOn + 1
      IF SOUNDOn >= 5 THEN
        LPRINT ERR
      END
    END IF
    FOR S = 1 TO 2
      SOUND 650, 2
    NEXT S
    FOR s1 = 1 TO 2
      SOUND 1400, 2
    NEXT s1
  NEXT x
  CALL zeroCTM51(BSA%)      ' Zero Counters
  CALL readCTM51(ct(), BSA%)
  RESUME

  CASE 24, 25
    CLS
    COLOR 7, 0
    LOCATE 23, 20
    PRINT "Turn Printer On - Press any key....."
    pause$ = INPUT$(1)
    CLS
    COLOR 7, 0
    RESUME

  CASE 27
```

```

CLS
LOCATE 23, 20
COLOR 0, 7
PRINT "Printer is out of paper - Press any key....."
pause$ = INPUT$(1)
CLS
COLOR 7, 0
RESUME

CASE 51, 57
CLS
LOCATE 10, 28
COLOR 0, 7
PRINT "Internal Error - Program aborted"
LOCATE 12, 28: PRINT "Press any key....."
COLOR 7, 0
pause$ = INPUT$(1)
CLS
GOTO EndPhase2
RESUME

CASE 53
CLS
LOCATE 10, 20
PRINT "File Not Found - Press Any Key To Continue"
SLEEP
IF previous = 1 THEN GOTO mainmenu
RESUME NEXT

CASE 75
CLS
LOCATE 10, 20
PRINT "Path/File Access Error - Press Any Key To Continue"
pause$ = INPUT$(1)
RESUME mainmenu

CASE 76
CLS
LOCATE 10, 20
PRINT "Path Not Found - Press Any Key To Continue"
pause$ = INPUT$(1)
RESUME mainmenu

CASE 11
RESUME NEXT

CASE ELSE
LPRINT
LPRINT "Error number "; ERR; " occurred"
LPRINT
RESUME NEXT

```

END SELECT

PrintHeader:

```
LPRINT
LPRINT CHR$(27); CHR$(71); CHR$(27); CHR$(14);
LPRINT TAB(30); UCASE$(stage$)
LPRINT CHR$(27); CHR$(87);
LPRINT
LPRINT "File - "; file$
LPRINT DATE$
LPRINT TIME$
LPRINT
LPRINT CHR$(27); CHR$(72);
RETURN
```

quit:

```
LPRINT CHR$(27); "P";
LPRINT CHR$(24);
LPRINT CHR$(27); "#";
LPRINT CHR$(27); "@";
CLS : CLOSE : END
```

AvailableVIS:

DATA 0, 4, 6, 8, 10, 12, 16, 20, 22, 23, 24, 26, 30, 36, 55, 144

FrequencyTables:

DATA "580","1230","2460","4485","7520"

AmplitudeTables:

DATA "1.6","1.7","1.9","2","2.1","2.3","2.4","2.5","2.6"
DATA "2.7","2.8","2.9","3","3.1","3.2","3.3","3.4","3.5"

DATA "1.0","1.2","1.4","C","1.8","2.0","2.1","2.2","2.3"
DATA "2.4","C","2.5","2.6","C","2.7","2.8","C","2.8"

' Voltage Divider must be used for following values....

DATA ".075",".085","C",".125",".195",".305","C",".315",".325",".445"
DATA "C",".555",".575",".775","C",".915","1.075","1.185","C","1.385"
DATA "1.675","1.915","C","2.225","2.505","2.935","C","3.345","3.895"
DATA "4.505","C","5.205","5.965","7.225","C","8.125","9.535","11.465"
DATA "C","13.595","16.535","19.975"

VItables:

' VI Tables for 0, 4, 6, 8, 10, 12, 16, 20, 22, 23, 24, 26, 30, 36

DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

DATA 0,0,0,0,0,1,1,1,1,2,2,2,2
DATA 3,3,3,4,4,5,5,6,6,7,8,9,12,17

DATA 3,4,4,4,4,4,4,5,5,5,5,5,6,6,6,6,7,7,8,9,9
DATA 10,11,11,12,0

DATA 3,3,3,4,4,4,5,5,6,6,6,7,8
DATA 8,9,9,10,10,11,11,11,12,12,12,12,13,13,14

DATA 3,3,3,4,4,4,5,5,6,6,6,7,7,10
DATA 10,13,13,14,14,14,15,15,15,16,16,16,17,17,17

DATA 3,3,4,4,5,5,6,6,7,7,9,9,12,12,12
DATA 12,12,12,15,15,17,17,18,18,19,19,20,20,21,21

DATA 3,3,4,4,5,5,7,7,9,9,11,11,12,12,16
DATA 16,20,20,21,21,23,23,25,25,27,27,28,28,29,29

DATA 3,3,5,5,7,7,9,9,10,10,12,12,14,16,20
DATA 20,24,26,28,28,30,30,31,31,33,33,35,35,37,37

DATA 3,4,4,5,5,7,8,8,9,10,10,11,12,16,22
DATA 22,28,32,33,34,34,35,36,36,37,39,39,40,40,41

DATA 3,4,4,5,6,6,7,8,9,9,10,11,15,18,23
DATA 23,28,31,35,36,37,37,38,39,40,40,41,42,42,43

DATA 3,3,4,5,7,8,10,10,12,12,14,16,18,22,24
DATA 24,26,30,32,34,36,36,38,38,40,41,43,44,45,45

DATA 3,5,5,7,7,9,10,10,12,14,16,18,22,24,26
DATA 26,28,30,34,36,38,40,42,42,43,45,45,47,47,49

DATA 3,4,5,7,9,9,11,12,14,16,18,22,26,28,30
DATA 30,32,36,40,42,44,46,48,49,51,51,53,55,56,57

DATA 3,5,8,10,12,14,16,18,22,24,26,30,32,34,36
DATA 36,36,38,42,44,46,50,54,56,58,60,62,64,67,69

DATA 4,6,8,11,13,15,18,20,23,26,29,32,36,39,43,45,48,52
DATA 57,63,69,76,84,93,104,118,137,165,180,242

DATA 2,7,12,17,23,29,35,41,47,54,62,69,77,86,95
DATA 104,115,126,138,151,165,181,199,220,244,273,309,358,434,633

DATA 19.975,16.535,13.575,11.465,9.535,8.125,7.225,5.965,5.205
DATA 4.505,3.895,3.345,2.935,2.505,2.225,1.915,1.675,1.385,1.185
DATA 1.075,.915,.775,.575,.555,.445,.325,.315,.305,.195,.125,.085,.075

DATA 3.933,4.516,5.185,5.953,6.835,7.848,9.011,10.346,11.878,13.638
DATA 15.659,17.979,20.642,23.700,27.212,31.243,35.872,41.187,47.228
DATA 54.294,62.338,71.574,82.178,94.353,108.332,124.381,142.809
DATA 163.967,188.259,216.15,248.173,284.941

SUB loadvi

LOCATE 10, 27

PRINT "Loading VI Tables"

FOR var = 0 TO 144 ' Load VI Schedules

FOR x = 1 TO 30

READ VI(var, x)

NEXT x

IF var = 0 THEN var = 3: GOTO rnv

IF var = 4 THEN var = 5: GOTO rnv

IF var = 6 THEN var = 7: GOTO rnv

IF var = 8 THEN var = 9: GOTO rnv

IF var = 10 THEN var = 11: GOTO rnv

IF var = 12 THEN var = 15: GOTO rnv

IF var = 16 THEN var = 19: GOTO rnv

IF var = 20 THEN var = 21: GOTO rnv

IF var = 22 THEN GOTO rnv

IF var = 23 THEN GOTO rnv

IF var = 24 THEN var = 25: GOTO rnv

IF var = 26 THEN var = 29: GOTO rnv

IF var = 30 THEN var = 35: GOTO rnv

IF var = 36 THEN var = 54: GOTO rnv

IF var = 55 THEN var = 143: GOTO rnv

rnv:

NEXT var

CLS

END SUB

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 92		3. REPORT TYPE AND DATES COVERED Interim
4. TITLE AND SUBTITLE Computer Software and Hardware to Determine Contrast Sensitivity Using Three Methods: Tracking, Limits, or Constant Stimuli			5. FUNDING NUMBERS 62758N MM 58524.002-0010	
6. AUTHOR(S) D.J. Hatcher, T.L. DeVietti, and J.A. D'Andrea				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NAVAEROMEDRSCHLAB 51 HOVEY RD PENSACOLA, FL 32508-1046			8. PERFORMING ORGANIZATION REPORT NUMBER NAMRL Technical Memorandum 92-4	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Naval Medical Research and Development Command National Medical Center Building 1, Tower 12 8902 Wisconsin Avenue Bethesda, MD 20889-5044			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) The software and hardware to obtain contrast sensitivity functions in a single session with monkeys are described. The versatility of the software allows the program to be used in a variety of sensory systems with minor programming modifications. It will permit absolute or difference threshold measurements with any one of three methods of threshold testing: (1) method of tracking, (2) method of limits, and (3) method of constant stimuli.				
14. SUBJECT TERMS Contrast sensitivity, Computer Program			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR	